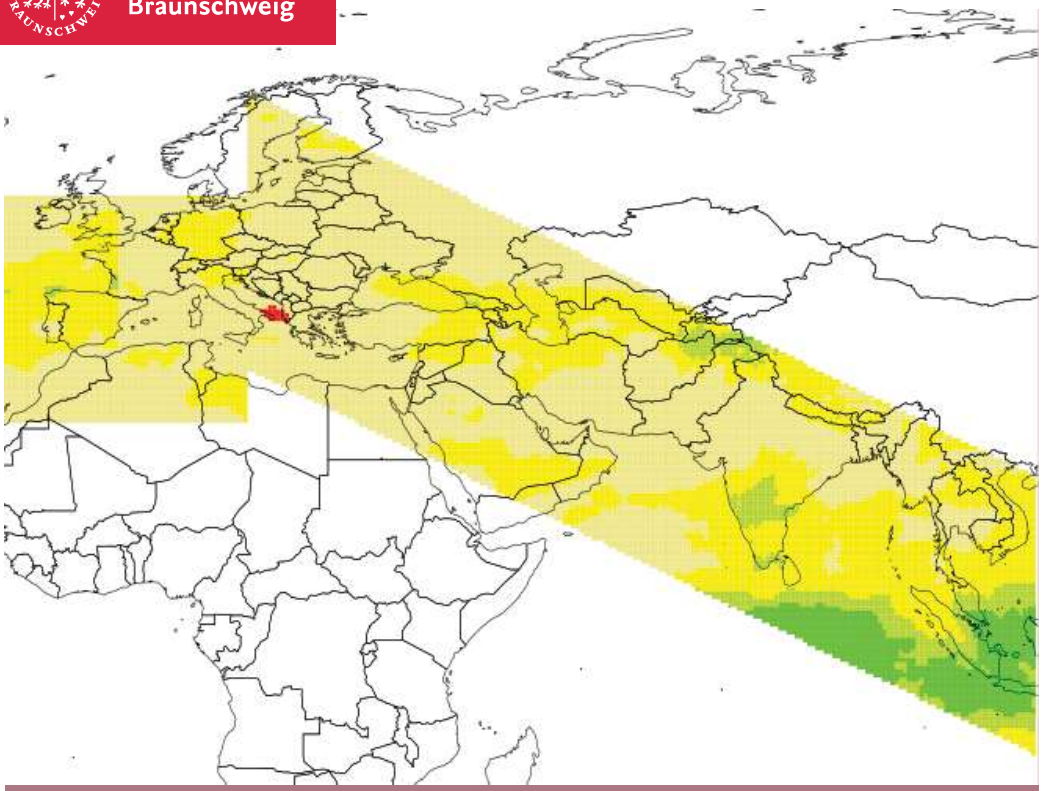
Technische Universität Braunschweig

NFL
am Campus Forschungsflughafen



Big Data Machine Learning for Flight Planning

Ralf René Shu-Zhong Cabos
TU Braunschweig, Institut für Flugführung

Niedersächsisches Forschungszentrum für Luftfahrt - Forschungsbericht 2018-08

TU Braunschweig–Niedersächsisches Forschungszentrum für Luftfahrt

Berichte aus der Luft- und Raumfahrttechnik

Forschungsbericht 2018-08

Big Data Machine Learning for Flight Planning

Ralf René Shu-Zhong Cabos

TU Braunschweig
Institut für Flugführung

Diese Arbeit erscheint gleichzeitig als von der Fakultät für Maschinenbau der Technischen Universität Carolo-Wilhelmina zu Braunschweig zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigte Dissertation.

Die Deutsche Bibliothek - CIP Einheitsaufnahme
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.de abrufbar.

Ralf René Shu-Zhong Cabos
Big Data Machine Learning for Flight Planning

ISBN 978-3-947623-04-4

als Manuskript gedruckt

Diese Arbeit erscheint gleichzeitig als von der Fakultät für Maschinenbau der Technischen Universität Carolo-Wilhelmina zu Braunschweig zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigte Dissertation.

Herausgeber der NFL Forschungsberichte:
TU Braunschweig–Niedersächsisches
Forschungszentrum für Luftfahrt
Hermann-Blenk-Straße 27 • 38108 Braunschweig
Tel: 0531-391-9822 • Fax: 0531-391-9804
Mail: nfl@tu-braunschweig.de
Internet: www.tu-braunschweig.de/nfl

Big Data Machine Learning for Flight Planning

Von der Fakultät für Maschinenbau
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung der Würde

eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Dissertation

von:	Ralf René Shu-Zhong Cabos, M.Sc.
geboren in:	Frankfurt am Main

eingereicht am:	07. Mai 2018
mündliche Prüfung am:	24. September 2018

Vorsitz:	Prof. Dr.-Ing. Peter Horst
Gutachter:	Prof. Dr.-Ing. Peter Hecker
	Prof. Dr.-Ing. Uwe Klingauf
	Dr.-Ing. Jens Schiefele

2018

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Flugführung der TU Braunschweig. An dieser Stelle möchte ich meinen Dank an all diejenigen aussprechen, die mich in dieser Zeit zur Erstellung dieser Dissertation begleitet und unterstützt haben.

Allen voran danke ich Prof. Hecker für die Betreuung der Arbeit und der zahlreichen Fachgespräche. Dank gebührt ebenfalls den Professoren Klingauf und Horst, welche sich als Gutachter bzw. Vorsitzender bereit erklärten.
Besondere Unterstützung erhielt ich stets seitens der Jeppesen GmbH und darüber hinaus von Boeing Research & Technology–Europe. Dabei geht ein Dank an Jens Schiefele für die Möglichkeit, diese Arbeit in enger Kooperation mit einem Industriepartner zu erstellen. Ein besonderer Dank geht an meinen Betreuer Nils Kneuper, der mich während der Arbeit inhaltlich stets unterstützte. Hervorzuheben sind dabei die zahlreichen Diskussionen, in denen wir beide jeweils bemüht waren, stets das letzte Wort zu haben. Diese Debatten führten immer zu neuen und wertvollen Ideen, wovon einige auch in dieser Arbeit erscheinen.
Für die Möglichkeit der Verwendung eines Datenclusters von Boeing möchte ich Miguel Vilaplana und seiner Arbeitsgruppe danken. Insbesondere David Scarlatti trug mit seiner Begeisterung für das Forschungsthema und seinem technischen Know-how maßgebend am Anfang der Promotion zu raschen Fortschritten bei.

Nicht zuletzt geht ein herzlicher Dank an meine Eltern, Geschwister, Großmutter und meine Verlobte Gianna für die ununterbrochene moralische Unterstützung über die letzten Jahre. In den doch zahlreichen Momenten, in denen ich meinen Kopf in den Sand stecken wollte, war es immer wieder Gianna, die mir wieder auf die Beine half.

Braunschweig, im Mai 2018

Zusammenfassung

Wettervorhersagen stellen einen wesentlichen Einsatz im Flugplanungsprozess dar. Während andere Eingangsgrößen meist mit einer hohen Gewissheit den Fluggesellschaften bekannt sind, besitzen Wettervorhersagen ein inhärentes Maß an Unsicherheit. Flugplanungssysteme haben jedoch in den meisten Fällen keine andere Wahl, als die Vorhersage als vollständig akkurat anzunehmen. Solche Unsicherheiten führen dazu, dass eine Trajektorie geplant wird, welche nicht die kosteneffizienteste ist. Tatsächlich wurden Vorhersageunsicherheiten als die größte Ursache von Trajektorienvorhersagefehlern identifiziert.
Wettervorhersagen werden allgemein durch numerische Simulationen der globalen Atmosphäre erzeugt.
Diese Simulationen basieren dabei auf Modellen, welche die physikalischen Prozesse modellieren. Da diese jedoch nur eine Approximation der komplexen Vorgänge in der Atmosphäre darstellen, entsprechen die Vorhersagen nicht der Wahrheit. Technologische Fortschritte im Bereich der Datenverarbeitung haben zu einer breiteren Fülle an Möglichkeiten zur Verarbeitung von großen Datenmengen, allgemein als *Big Data* bekannt, geführt. Diese Datenverabeitung schließt Datenanalysen und Methoden des maschinellen Lernens ein, womit für das menschliche Auge nicht bekannte Muster bzw. Funktionszusammenhänge detektiert werden können. Es ist unerforscht, ob Unsicherheiten in Wettervorhersagen ebenfalls mit diesen Methoden prognostiziert werden können und ob damit ein Vorteil auf den Flugplanungsprozess erzeugt werden kann.

Diese Arbeit liefert eine Durchführbarkeitsuntersuchung einer datenbasierten Herangehensweise an die Prognose von Vorhersageunsicherheiten. Im Anschluss erfolgt eine Validierung potentieller Vorteile auf die Planbarkeit eines Flugplanungssystems.
Dafür wird ein Datencluster benutzt, worauf globale Wettervorhersage- und Re-Analyse-Daten über knapp unter zehn Jahre aufbereitet werden. Acht Algorithmen des maschinellen Lernens werden anhand dieser Daten mit der Diskrepanz zwischen besagten Datensätzen trainiert. Ziel ist es, dass die Algorithmen zugrundeliegende Muster der Unsicherheit erlernen. Dieses erlernte Wissen kann anschließend anhand eines Testdatensatzes auf die algorithmische Vorhersageleistung geprüft werden. Gleichzeitig kann so auch der am besten prognostizierende

Algorithmus je Vorhersageinstanz bestimmt werden. Eine zweite algorithmische Schicht wird im Anschluss realisiert, welche diese Testergebnisse zur Bestimmung des womöglich am besten prognostizierenden Algorithmus je Vorhersageinstanz eines weiteren Datensatzes benutzt.
Dieser Validierungsdatensatz erstreckt sich zeitlich über ein Jahr und liefert somit zeitgleich die Vorhersagen für die Flugplanung von drei Flugverbindungen. Diese Pläne werden im Anschluss mit den echt geflogenen Trajektorien verglichen, indem eine Diskrepanz ermittelt wird. Dadurch wird evaluiert, ob die Diskrepanz des mit Algorithmen prognostizierten Flugplans geringer ist als die des Plans basierend auf den Originalvorhersagen. Ergebnisse zeigen, dass die Algorithmen die Unsicherheit in einer Mehrheit von Fällen verringern können. Nachfolgende Ergebnisse zu den Flugplänen zeigen keinen ausschließlichen vorteilhaften Einfluss. Eine starke Korrelation zwischen Vorhersageleistung und durchflogener Weltregion kann dabei beobachtet werden.
So kann kein Vorteil bezüglich der Vorhersagbarkeit des Flugplanungssystems für einen Flug in Südostasien bestimmt werden, während ein Vorteil für einen interkontinentalen Langstreckenflug identifiziert wird.

Weitere Validierungen durch eine größere Zahl an Flugverbindungen über alle Weltregionen zur Bestimmung der Machbarkeits- und Zuverlässigkeitsgrenzen sind nötig. Weiterführende Forschung wird zum Verständnis zugrundeliegender Muster sowie zur Erhöhung der Systemzuverlässigkeit benötigt.

Abstract

Weather forecasts serve as a fundamentally important input to the flight planning process. While other types of input data are mostly known to airlines to a high degree of certainty, weather forecasts carry an inherent measure of uncertainty. Flight planning engines commonly have no other means but to consider a forecast to be entirely accurate. Such uncertainties thus lead to a trajectory being planned that does not represent the most cost-optimal option. In fact, weather forecast uncertainties have been identified to be the greatest source of trajectory prediction errors. Weather forecast generation relies on numerical simulations of the earth's atmosphere, which in turn rely on models imitating the physical processes involved. However, these represent approximations of reality and are thus not able to perfectly capture the complex processes involved. Technological advances have meanwhile lead to a surge in means to more efficiently process large amounts of data, commonly termed *Big Data*. Such processing includes the possibility of applying analysis and Machine Learning techniques, in order to apprehend any patterns otherwise undetectable to the human observer. It is therefore of interest whether forecast uncertainties can be predicted using these means and whether these predictions in turn yield a benefit for the flight planning process.

This thesis provides a feasibility evaluation of a data-centric approach to weather forecast uncertainty prediction and a subsequent validation of potential benefits to a flight planning engine's measure of predictability. Core to this research is a data cluster, on which global weather forecast and re-analysis data spanning close to ten years have been gathered. Eight Machine Learning algorithms are trained on this data using the discrepancy between forecast and re-analysis data. Doing so ensures that the algorithms learn an underlying pattern of forecast errors or uncertainties. This can in turn be utilized to predict the uncertainties in a test set to determine the best-performing algorithm per forecast instance. A second algorithmic layer is further realized which leverages this information to determine the algorithm generating the most accurate prediction, per forecast instance. A validation data set spanning a year of data is utilized to serve as input data for the flight plan generation of three flights. These are then compared to the flight's actual flown trajectories. It is examined whether the discrepancy between flight plan and trajectory is decreased with a flight plan based on predictions of the methodology

herein, as compared to a control. Results indicate that algorithms' predictions are able to decrease forecast uncertainty in a majority of cases. Subsequent flight plan results indicate an ambivalent result. A heavy dependence on the world region the flight is performed in, is recorded. As such, no benefit to flight plan predictability is observed for a short haul flight in South East Asia, while a slight benefit is recorded for an intercontinental long haul flight.

An operational realization is not recommended at the time of writing, as further validations covering more areas and a greater number of flights need to be performed to better gauge the boundaries in which the method is beneficial to the flight planning process. Further research is needed to understand the underlying patterns in algorithmic prediction performance and increase reliability.

Contents

List of acronyms

2D 2-dimensional

3D 3-dimensional

4D 4-dimensional

ACARS Aircraft Communications Addressing and Reporting System

ACMs Atmospheric Circulation Models

ADS-B Automatic Dependent Surveillance - Broadcast

AI Artificial Intelligence

ATC Air Traffic Control

ATM Air Traffic Management

CFSR Climate Forecast System Re-analysis

CFSv2 Climate Forecast System version 2

CI Cost Index

CMC Canadian Meteorological Centre

CPU Core Processing Unit

CSV Comma-separated Values

DIVMET Adverse weather diversion model

ECMWF European Centre for Medium-Range Weather Forecasts

EPS Ensemble Prediction System

EUROCONTROL European Organisation for the Safety of Air Navigation

ETD Estimated time of departure

ETL Extract, transform, load

EWR Newark Liberty International Airport

FIR Flight Information Region

GB Gigabytes

GFS Global Forecast System

gpm Geopotential meter

GRIB General Regularly-distributed Information in Binary form

GS Ground Speed

HDFS Hadoop Distributed File System

HGT Geopotential height

ICAO International Civil Aviation Organization

IFR Instrument Flight Rules

ISA International Standard Atmosphere

JMA Japan Meteorological Agency

KDD Knowledge Discovery from Data

kNN k-Nearest-Neighbors

KPI Key Performance Indicator

mbar Millibars

MRO Maintenance, Repair and Overhaul

MSE Mean squared error

NAVAIDS Navigation Aids

NCEP National Centers for Environmental Prediction

NextGen Next Generation Air Transportation System

NoSQL not-only-SQL

NWP Numerical weather prediction

NWS United States National Weather Service

OR Operations Research

PDF Probability Density Function

QP Quadratic Programming

RAD Route Availability Document

RAM Random Access Memory

RBF Radial Basis Function

RDBMS Relational Database Management Systems

RNP Required Navigation Performance

RSS Residual Sum of Squares

RUC Rapid Update Cycle

RVSM Required Vertical Separation Minimum

SESAR Single European Sky ATM Research

SD Standard Deviation

SIN Singapore Changi International Airport

SVM Support Vector Machine

TAS True Air Speed

TB Terabyte

TBOs Trajectory-based Operations

TMP Temperature

TOC Top of Climb

TOD Top of Descent

UGRD U wind component

UI User Interface

UKMO United Kingdom Meteorological Office

U.S. United States

UTC Coordinated Universal Time

VGRD V wind component

WMO World Meteorological Organization

1 Introduction

This chapter serves as an introduction to the topic of Big Data Machine Learning in Flight Planning, by first providing the motivation behind the conceptualization and realization of such a system. Afterwards, the goals, the approach, and the dissertation's structure are presented.

1.1. Motivation

Weather forecasting predictability is generally dependent on estimating the uncertainties of both initial conditions and the propagation of these as a result of the chaotic nature of weather [1]. Forecasting relies on numerical simulations and corresponding models imitating the atmosphere's physical properties. These models, such as Atmospheric Circulation Models (ACMs), generate forecasts across areas of wide geographical coverage [2]. Since these models cannot fully capture the atmosphere's complex physical processes, the resulting forecasts will exhibit uncertainties.

Efforts in the meteorological realm have been undertaken which aim to capture a measure of uncertainty in deterministic weather models. These rely mainly on creating a number of forecasts based on varying starting conditions of the numerical simulation [3, 4]. By doing so, a quantitative spread of forecasts is created which serves as a measure of likely forecast deviation. To date, no system for forecast uncertainty prediction entirely relying on large amounts of data and machine learning methods has been conceptualized and tested.

These uncertainties pose a challenge to airline flight planning and the wider Air Traffic Management (ATM) system as a whole. Flight planning represents a mandatory process for airlines under International Civil Aviation Organization (ICAO) Doc 9976 [5], which airlines submit flight plans to the ATM system. While mandatory, this process also opens a window of possible cost-efficiency. Weather cannot be ignored in this context, hence the requirement for weather forecasts.

A flight planning engine will in any case generate the most cost-efficient route, given the input data. There is no alternative than to assume that all data is entirely accurate. Any uncertainties will be translated into the resulting flight plan. Beside the cost factor, a flight plan's other important aspect is its predictability. Ideally, for both the ATM system as well as airlines, any flight is handled exactly per its plan.

In reality, different inputs to the flight plan engine carry along uncertainties which result in a discrepancy between flight plan and trajectory. Cole et al. [6] showed that weather accounts for the greatest inaccuracies in trajectory prediction.
Over the last decade, methods and systems to handle and process large amounts of data have matured. Technologies such as the Apache Foundation's *Hadoop* allow the creation and handling of large batches of data [7], while frameworks such as *Spark* allow efficient parallel computing [8]. This trend is mirrored by the drive in recent years to extract information from large batches of data in order to achieve an operational benefit.
This dissertation's motivation connects this stated drive towards data analytics with the topic of forecast uncertainties. Specifically, the research question set forth in this dissertation is whether data analytics can be applied in forecast uncertainty estimation and whether any achieved accuracy gains are translated to an increase in flight plan predictability.

1.2. Goals

In light of the application to flight planning, the goal of this thesis is to conceptualize, realize and evaluate a forecast uncertainty prediction system. For this purpose, three goals are stated to be completed in the course of this dissertation:

- **Machine learning:** Reliance on prediction generation solely on the basis of machine learning algorithms. Instead of approximating physical processes, these algorithms are to identify patterns of uncertainty in the data provided.
- **Data compatibility:** Derived from flight requirements, the resulting data analysis is required to be performed across large geographical areas and throughout a number of time steps.
- **Retrofitting capability:** The system is required to be designed in a way that if determined operationally feasible, a integration into current flight planning processes is easily realizable.

The Big Data machine learning system is conceptualized and realized while achieving the above-stated goals. This is followed by an evaluation regarding the algorithmic prediction performance and the subsequent effect on the flight planning process' predictability.

1.3. Approach

The approach undertaken in this dissertation consists of a data cleaning and handling step. Specifically, this includes the upload of weather data covering a time

span of 9 years and 8 months to a Big Data cluster and the subsequent coercion into a format suited for distributed processing. Forecasts act as the input, while re-analysis data is utilized as the target function, i.e. the *true* weather value. Data handling steps provide the groundwork to allow parallel processing by grouping the data set into relevant batches. A split into training and test subsets is performed per batch. On these, a selected number of algorithms are trained and subsequently tested for their prediction performance.
These test results are to be stored in the data cluster and utilized for a second algorithmic layer, called the *algorithm selection method*. This layer consists mainly of a k-Nearest-Neighbors k-Nearest-Neighbors (kNN) algorithm, which selects the most accurate algorithm per forecast instance, based on the test results. A second data set covering a year of data is then to be used as a validation data set. The data herein acts as an input to the selection method, which in turn determines the algorithm likely to yield the most accurate prediction. The output of this step is once more determined per its prediction performance.

These predictions are eventually to be used to generate flight plans. For this purpose, one based on the original forecast data and a second one based on the output predictions is to be generated. With the corresponding actual flown trajectory, the discrepancies of both flight plans are to be calculated and compared to one another. Envisioned is the determination as to whether the discrepancy of the flight plan generated with the algorithms' output is lesser than that of the flight plan calculated with the original forecast. Doing so creates a verdict as to whether predicting weather forecast uncertainty increases flight plan predictability.

1.4. Structure

This dissertation is structured into eight chapters and one Appendix containing results of the various chapters on evaluations. The following list provides a short description of each chapter:

- **Chapter 2** provides insight on the topics of flight planning, Big Data analysis technologies (including machine learning) and weather forecasting. It concludes with a description of a conceptual proposal for a machine learning system for uncertainty prediction.
- **Chapter 3** defines general requirements derived from the flight planning process and proposes a high-level system, while detailing the specific data needs associated with this.
- **Chapter 4** details the data cleaning and handling process, including the steps

necessary for coercion into a data format suited for efficient parallel computing.

- **Chapter 5** describes the development of the algorithmic training and selection scheme. Considerations as to computational complexities as well as the implementation in the data cluster is provided.
- **Chapter 6** evaluates and discusses the results obtained from the testing of algorithms, as well as the algorithm selection method. Lastly, a sensitivity analysis aimed at finding the most optimal setting for the selection method is outlined and the results of which discussed.
- **Chapter 7** presents the approach undertaken to evaluate the concept's feasibility in the context of flight planning. A process of comparing a number of different types of flight plans to the actual trajectory is outlined and the results discussed.
- **Chapter 8** concludes this dissertation by providing a summary of the entire work. Furthermore, potential improvements are outlined in an outlook, by which the concept's effectiveness may be increased. This outlook also includes a hypothetical implementation in an operational setting.

2 Fundamentals and State of the Art

This chapter describes the underlying fundamentals, as well as the State of the Art for a number of competency areas relevant to the field of research. For this, three general areas with their underlying theory are first motivated, presented and discussed. By doing so, a common layer of understanding can be established, on which the research question can be founded. The three areas are:

1. Flight planning
2. Big Data analysis technologies
3. Weather forecasting and forecast verification

The fourth and final part of this chapter focuses on the joining of these three areas and the subsequent identification of the gap in the current state of research. This gap is the current lack of application of machine learning methods to predict forecast uncertainties.

2.1. Flight planning

The goal of flight planning is to ensure that a flight adheres to operational regulations and that the flight crew receive all required information, in order to safely conduct the flight while coordinating all actions with Air Traffic Control (ATC) [9]. A flight plan's details vary between regulators, but always involve those relating to the insurance that sufficient fuel is carried for the planned route and foreseeable irregularities [10]. Most commonly, details such as the departure time, route, altitude, speed and aircraft type being used are included in the flight plan [11]. Airlines utilize software to calculate flight plans, in which proposed operations are compared to regulatory constraints. Legal documents are produced by the software, to confirm that the planned operation is legal, i.e. does not violate any regulations. In large parts of the world flight planning is performed by dispatchers, while in Europe, the legal requirement does not call for such staff [10].
From a regulatory viewpoint, flight planning represents one part of preparations of

any commercial flight, as defined in section 4.3.3 of *Annex 6, Part I* by International Civil Aviation Organization (ICAO) [12]. The process is mandatory for all commercial flights and is defined by a set of standards by ICAO's *Doc 4444, Procedures for Air Navigation Services - Air Traffic Management* [13]. Requirements and instructions are defined to ensure that operators adhere to certain requirements prior to departure, such as Required Navigation Performance (RNP) or Required Vertical Separation Minimum (RVSM), where applicable [13].
While regulatory constraints ensure operational safety, they effectively prevent airlines from entirely customizing their flight plan based on their respective operational goals. Within the regulatory constraints imposed, airlines are still given the opportunity to customize parts of the flight plan. It is the task of flight planning software to determine the most cost-effective solution under consideration of constraints and uncertainties. This makes flight planning inherently an Operations Research (OR) problem [10], in which a cost function is established, with the aim being to find the minimum of this function [14]. It is determined by the variation of variables, such as routes or speeds. These in turn are further bound by their domain aspects. [15]

2.1.1. Relevant domain aspects

The following three domain aspects serve as the operational boundaries in flight planning: aircraft performance, weather and route and altitude structure [15]. Any flight can only be planned within the limits set forth by these aspects. This section presents an introduction to each of the three.

Aircraft performance

Aircraft performance represents the first of the three domain aspects in flight planning. Different aircraft carry along varying fuel burns, for e.g. the climb section of a flight [10]. Fuel requirements vary due to aircraft weight, with heavier aircraft requiring more fuel than lighter ones for the same change in altitude. Differing values in fuel burn result in varying aircraft weight during flight, which in turn influences the optimal decision at any point during the flight [15]. The fuel burn per hour as well as the reachable distance at an arbitrary altitude and speed depends primarily on aircraft weight [15]. Based on flight mechanics equations [16], more optimal flight states concerning altitude and speed can be calculated and which need to be taken into account by any flight planning tool. Aspects on aircraft performance are further detailed by Padilla [17] and Airbus [18].

Weather

The flight characteristics depend heavily on the wind speeds and temperatures encountered [10]. Fuel flow is particularly dependent on the ambient air temperature (for an arbitrarily given weight, speed and altitude). The optimal route depends on the wind speeds encountered, as the Ground Speed (GS) is a simple vector addition of wind speed and the True Air Speed (TAS). A wind-optimal route can thus result in a distance up to 10% longer than the great circle route, albeit with lesser fuel burn [10]. Weather data is of great importance, with the current primary sources being the United States National Weather Service (NWS) and United Kingdom Meteorological Office (UKMO). Weather forecasts are considered to be deterministic and accurate by most flight planning systems. In contrast to aircraft performance, weather carries an element of uncertainty. This uncertainty in wind and temperature forecasts is in part the justification for contingency fuel [10], which serves as a means of compensation should *"unforeseen meteorological conditions"* result in a higher-than-planned fuel burn [5].

Route and altitude structure

The third domain aspect concerns air routes and the altitude structure. Static airways cover a large area of the world [10, 15]. These effectively represent an equivalent to a road network for automobiles on the ground. Static airways are defined by permanent points on the ground, of which some are defined by the location of a Navigation Aids (NAVAIDS), while others are merely described by coordinates. Distances between the points represent airways. Most commonly, it is forbidden to deviate from these airways. In some parts of the world, it is allowed to fly directly between points not connected to each other through an airway. These *point-to-point* connections, although limited in their numbers, are available in parts of the United States (U.S.), Canada and Scandinavia. [10, 15]

Additionally, a number of dynamic airways are available, the tracks and altitudes of which are published on a daily basis [15]. These airways are usually published for high traffic density areas, such as the North Atlantic. In some oceanic areas, aircraft are even permitted to fly unstructured routes between points. [10]

Route structures are evolving constantly. Adding to the non-triviality of the definition of the structure are regulatory restrictions, which are published on a regular basis and advice on availability of routes. [15] A prime example is the *Route Availability Document (RAD)* by the European Organisation for the Safety of Air Navigation (EUROCONTROL) [19], which is published on a 28-day basis. When planning a flight, the planning engine will therefore have to account for all RAD restrictions on top of those already imposed by static and dynamic routes. [15]

Future concepts, such as Single European Sky ATM Research (SESAR) or Next Generation Air Transportation System (NextGen) have at their core the transformation to Trajectory-based Operations (TBOs) [20]. This steps represents a shift away from the need to fly along certain airways and instead along the most direct route.

2.1.2. Problem formulation and cost function

A formulation of flight plan optimization has not been clearly postulated, with authors generally reluctant to present a clear definition of the problem [10]. The reason for this may lie in the fact that, seen from the viewpoint of an optimal control theory approach, the design of the problem's complexity quickly increases [21]. Extending this approach, McIntyre [22] presented an outcome to flight planning optimization should constraints not be sufficiently accounted for in a 3-dimensional (3D) network, with speed representing the fourth dimension.
Two approaches to the 4-dimensional (4D) route/trajectory optimization problem can thus be presented; *optimal control theory* and *network optimization*. These two approaches are presented concisely in the following sections. For the latter, a reference is drawn to 2.1.1 and the cost function of the optimization problem is defined.

Optimal control theory

The first approach to solve the 4D optimization problem is by applying optimal control theory. For this, methods based on prior work associated mainly with Bryson and Ho [23, 24], are utilized. An objective function, as in (2.1), is to be minimized, for which the equations of motion are written as a system of differential equations and numerically integrated. Control variables, such as the local speed and direction of flight are then optimized to achieve the required minimization of the function. [10]

$$\dot{X} = F(X, T, U, \phi) \tag{2.1}$$

In this formulation of the approach, X indicates the state vector of the aircraft's 3D position as well as its mass and F a function defining the system dynamics. Furthermore, the function is dependent on temperature T, wind speeds U and a further vector ϕ of controls of how the aircraft is operated. [10] The problem is further converted by the replacement of the equations of motion with constraints ensuring physical continuity. By avoiding the finite-difference solution through the replacement of X by a set of decision variables Z, the trajectory can be divided into k timesteps and thus, resulting in k additional constraints. This results in a large problem with a sparse constraint matrix, for which nonlinear programming algorithms can be applied. [10]
According to Karisch et al. [10], this approach's essential disadvantage is that the

produced paths are not domain feasible and hence need a *"...complicated correction cycle"* [21] for fitting the optimal to the allowable route and altitude structure.

Network optimization

An approach based on network optimization focuses on two necessities: the identification of the set of nodes and edges of the problem and the dynamic cost nature of an arbitrary path, under consideration of other decision variables and constraints [10]. Edges represent the network of airways already explained in 2.1.1. Using domains to establish the network results is a non-trivial information management problem, however is mathematically straightforward [10]. The first to propose an approach based on network optimization was de Jong [21], while employing the *shortest path algorithm* by Bellman [25].

Karisch et al. [10] offer a high-level mathematical formulation of the network optimization problem: all regulatory and operational constraints are first captured as rules. This determines the network itself and determines whether an edge can connect two nodes. The set of constraints Z, as described in 2.1.2, consist of the edges themselves as well as the state of the aircraft in question and the departure time t_D. The set of edges $E = \left\{e_j\right\}$ with $j = 1, ...B$ available are not pre-determinable and must therefore be evaluated against Z. This is due to the dependance on what edges are actually traversed or during what time an edge is reached, which is yet again dependent on t_D as well as the path taken thus far. Altitudes are furthermore attributable to each edge, as in $H_{e_j} = \left\{h_\gamma\right\}$ and are dependent on the time the edge is reached. With these definitions, four types of costs can be defined: fuel, time, overflight fees and lost revenue due to spilled payload. The objective of any flight planning tool is thus to retrieve the optimal 4D path from origin to destination in a 3D environment, where speeds V_i, $i = 1, ...n$ represent the fourth dimension. The cost function of the optimization problem can therefore be stated as in (2.2):

$$\textbf{Min}\ C_F(W_D - W_A) + C_D(t_A - t_D) + C_A(t_A - t'_A) + \sum_{r \in FIR} C_N(r) + \sum_{k \in K} C_P(P_k) \tag{2.2}$$

Table 2.1 lists and names the terms included in the cost function, with t_A and t'_A as the actual and scheduled time of arrival and r and k as indexes for the specific Flight Information Region (FIR) and payload element spilled. A more detailed elaboration on each cost term is provided in [10], including an elaboration of the Cost Index (CI) [26].

Cost term	Type of cost
$C_F(W_D - W_A)$	Fuel cost; with C_F as cost per weight unit and weights at departure and arrival
$C_D(t_A - t_D) + C_A(t_A - t'_A)$	Times-based costs; divided between costs based on duration of flight (C_D) and difference between scheduled and actual arrival time (C_A)
$\sum_{r \in FIR} C_N(r)$	Overflight charge costs (C_N); often divided between costs based on the route flown and a flat fee for entering the FIR
$\sum_{k \in K} C_P(P_k)$	Spill costs (C_P); refers to lost revenue when payload is substituted for fuel (to increase range)

Table 2.1.: List of terms of the flight planning optimization cost function, after Karisch et al. [10].

2.1.3. Constraints

A number of constraints have to be taken into account when solving the cost function. These represent the operational boundaries of the domain and can generally be grouped into three categories: fuel and time of flight constraints, weather, maximum speed and altitude and contingency fuel constraints [10]. Fuel and flight time constraints aim to represent the restrictions for the consumption of fuel at an arbitrary point in flight, as without knowledge and calculation of the flight up until that point, the fuel flow until then cannot be determined. Weight limitations are another important aspect when looking at aircraft performance, as well as the distance able to be flown with an arbitrary amount of fuel. Weather, through wind and temperature, features two heavily influential constraints on the optimal flight route. Fuel flow is dependent on the fluid's density, which in turn is a function of temperature. On the other hand, GS is a vector addition of the wind speed and TAS. Accurate weather forecasting is of high importance to the accuracy of flight planning. Maximum speed and altitude constraints are also needed to account for feasibility when choosing an altitude. The optimal altitude and speed along each edge is dependent on the current weight of the aircraft, thus requiring knowledge of the flight's progress up until that point. The last constraint is that of the amount of extra fuel to be carried under ICAO regulations [5] to account for deviations from the planned path, in both distance and time. [10]

2.1.4. Solution approaches

Solution of the network optimization approach described in 2.1.2 is challenging when looking at the computation necessary [10]. Two approaches have to date been pursued, for which a brief description is presented in the following:

Sequential 2-dimensional (2D) network solution approach

This approach avoids the complex computation necessity by splitting the problem into two 2D subproblems: route optimization and profile/speed optimization. [10] The first block solves the ground track, while the second calculates the altitudes, speeds, payload and departure fuel. Constraints from Air Traffic Management (ATM), traffic, severe weather and aircraft capabilities are utilized for the route optimization calculation, with forecast winds used to determine each edge's wind component. The latter represents the fundamental difficulty when applying this approach, as winds vary by altitude. It is necessary to establish a heuristic rule, with which convergence towards an optimal solution can be determined. This applies for cruise speeds, as the optimal cruise speed is dependent on the wind incurred. [10]

4D network solution approach

This approach aims to extend the foregone 2D approach by adding nodes at all available altitudes for every lateral waypoint. In this way, a 4D problem is created, to be solved directly. While such an approach poses higher complexity and significantly more computation effort, the utilization of so-called "smart" heuristics can yield shorter computation times and in parallel, avoiding a major deterioration of the quality of the solution. [10] Further details on this approach can be found in [21], as well as by Sorensen and Goka [27] and Wilson et al. [28], who both show applications of this approach.

2.2. Big Data analysis technologies

The amount of data being generated in recent years has increased tremendously in every industry, including aviation [29, 30]. Concurrently, parallel and distributed computing has matured. These have given incentives to industries to focus on utilizing so-called *Big Data* technologies in order to gain a technological advantage. [29] While there is no single definition of *Big Data* (due to the relatively recent creation of the term in the past two decades), a commonly stated one is of the *Three V's*, first described by Laney [31]. The three V's stand for *volume* (size of data and storage space), *velocity* (speed of data arriving or being generated) and *variety* (types of data generated). While this definition provides no tangible boundaries when data can

be defined as Big Data, a common consensus is that data sets are defined as such when they cannot be efficiently handled with current data analysis and processing tools [32].
Stakeholders in the aviation industry, such as airlines, airports or aircraft manufacturers have long depended on data to plan their operations [32]. The diversity of data is yet so complex that manual analysis is infeasible. Big Data analysis tools, as well as the architecture needed to support these, have thus gained attention and could handle the formerly manual analysis in a more feasible way [32].

2.2.1. Big Data architecture

Big Data analysis tools tasked with the retrieval of information require an infrastructure out of the ordinary that is able to handle the vast amounts of data [32]. This is especially important when data sets exceed the size of 10 Terabyte (TB). At this data size, most Relational Database Management Systems (RDBMS) are not able to cope and handling of the data becomes difficult [32]. The industry has developed systems and solutions which deal with the challenges of Big Data, namely the size and the unstructured nature of data sets [33]. A common solution that is widely used throughout the industry, relies on Apache's *Hadoop* technology [33]. This distributed computing framework provides large-scale processing on cheap commodity hardware [29]. On top of this "foundation", a number of tools can be connected to support processing, such as *HBase*, *Hive*, *Pig* or *Spark* [34, 29]. Such a commonly-used *Hadoop stack* architecture is schematically illustrated in fig. 2.1. The core element for any Hadoop system is the Hadoop Distributed File

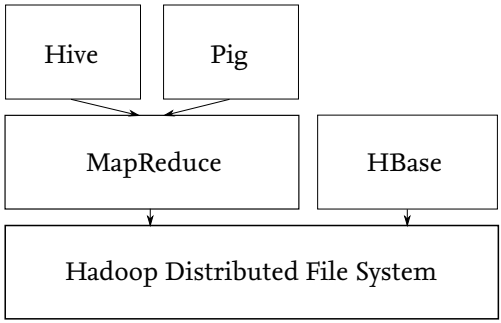


Figure 2.1.: The general architecture of a common Hadoop *stack*, or *ecosystem*, after Farooqi [34].

System (HDFS), which provides the storage capacity for large datasets [35]. While

providing the possibility of scaling capacity further and storing files in any format, the HDFS's advantage lies in the possibility of running *MapReduce* jobs [29]. Due to the distributed nature of Hadoop, storage and thus processing has to be performed across multiple nodes [36]. MapReduce is essential to any Hadoop processing, as it simplifies the distributed data processing functions across all nodes [36]. A single data processing mechanism would not be able to complete this task efficiently [37]. By realizing processing on a multitude of nodes, the overall time required can be significantly shortened [34].
The amount or diversity of other tools connected in the Hadoop ecosystem is not limited. Some database types such as not-only-SQL (NoSQL) ones, may serve a better purpose, depending on the use case. The programs utilized for running analysis on Hadoop also vary, with common solutions being *Hive* or *Pig* [38]. A combination of different analysis tools can be beneficial, as each has its advantages in handling data. Hive is suited for handling structured, whereas Pig's strength lies with unstructured data sets [39].
There is no single solution to the needed Big Data architecture, which explains the different design approaches undertaken by Murugan et al. [29], Begoli and Horey [40], Boci and Thistlethwaite [41] and Ayhan et al. [42], among others. Bakshi [33] outlined a generic approach to the design of a Big Data architecture, including performance and capacity considerations.
The next paragraphs provide an introduction to three core components of the Hadoop ecosystem and which are mainly relied on in this thesis.

Hadoop Distributed File System The Global Forecast System (GFS) is available as open source software by the Apache Software Foundation under the name *Hadoop*. The structure of the HDFS is profoundly similar to that of the original file system.
The HDFS runs on a master-slave architecture, specifically: the namenode (being the master) and the datanodes (being the slaves) [7]. The namenode's job is maintaining of the filesystem tree as well as the metadata for files and directories. Two files, the namespace image and the edit log, store the this information on the local disk. Additionally, the namenode also knows the datanodes which feature all the blocks for a given file. A block in the HDFS features by default a storage capacity of 64 MB, although the usage of 128 MB blocks is common. [7]
Datanodes act as the workers of the file system. Their task is to store and retrieve blocks of data, as being told by the namenode. They also report in parallel to the namenode with lists about which blocks they are storing. This structure therefore indicates that the file system is dependent on the resilience to failure of the namenode. Hadoop provides two ways for ensuring this. One way is to create back-

ups of the files which make up the persistent state of the file system's metadata. Configuration allows the writing of the persistent state to multiple file systems in Hadoop. The second possibility is to run a second namenode. While it replicates the original one's functions, it does not replace the first one and is usually run on a separate physical machine due to a high CPU demand. [7]

MapReduce MapReduce is the common tool utilized in Hadoop ecosystems to simplify the running of distributed data processing functions across multiple nodes in a cluster [36]. Because the HDFS is based on potentially thousands of storage nodes, a single data processing mechanism is not realizable in an efficient way, considering the amount of data stored in the system [37]. Another factor is the envisioned short response time in data processing. By increasing the number of processors, the overall data process time is unarguably shortened [34]. The advantage MapReduce offers is that no knowledge of distributed programming is needed for the creation of parallel processing [36]. MapReduce works on the structure laid out in 2.2.1: First, the HDFS splits the to be processed file into evenly large segments. These are then distributed by MapReduce to namenodes, which in turn assign the *Map* task to their currently available datanodes. [36]

Spark Apache Spark is a framework providing the means for cluster computing [43, 8]. It manages the distribution of processing jobs in the cluster, by dividing a main large job into smaller parts. The processing effort per node is judged by the available resources, i.e. the number of nodes and Core Processing Unit (CPU) per node. Spark then ensures an even distribution throughout all available resources. Spark is an upgrade from its predecessor, MapReduce, mainly in processing speed. [8] This is due to its reliance on in-memory computing, contrasting MapReduce. Also, the processing scheme is structured differently. Instead of dividing the processing job into series of *Map* and *Reduce* steps, Spark first reads in the entire function that needs to be executed on the data set. By doing so, the total computing effort can be determined and job division performed appropriately. Spark's ar-

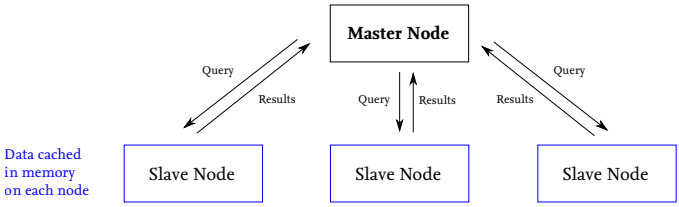


Figure 2.2.: Apache Spark working logic, with a single master node and multiple slave nodes, after Cabos et al. [44].

chitecture works mainly with two components: a master node and multiple slave nodes, as illustrated in fig. 2.2. The master node, holding the read in function, passes this function in the form a query to each slave node. Every slave node then executes this query on one part of the data set, which is cached in memory. After completion, the results are returned to the master node and aggregated and/or saved, either locally on the master node or in the HDFS. Due to Spark having to read the entire function prior to job execution, *lazy evaluation* is needed, with Spark only actually executing any function when a result is called for. This can be a save or write function, among others.
The distribution of a large job onto a number of nodes realizes the possibility of parallelizing parts of a job and thereby increasing the processing efficiency. This function legitimizes Spark's deployment in a cluster. Nevertheless, while yielding greater efficiency than MapReduce, both can be installed on the same cluster and do not interfere with each other.

2.2.2. Knowledge discovery

For the extraction of information from data, the term Knowledge Discovery from Data (KDD) is commonly used [40]. This process is oftentimes not limited to one field of science, but merges computer science, statistics, visualization as well as the understanding of the problem domain at hand. According to Begoli and Horey [40], KDD consists of three processes:

1. Collection, storage and organization of data
2. Application of analysis tools and methods
3. Thorough understanding and interpretation of the data

The architecture for the first point has been described in 2.2.1. This section focuses on the analysis tools and methods, with which knowledge or information can be extracted from data.

Terminologies The science of extracting information from data features a number of terminologies. Besides KDD, the term *data mining* is widely used, with both representing the same field [45, 46]. Other authors [40, 47, 48] define KDD to be on the conceptual level, with data mining being the technology applied. Data mining aims to retrieve useful or informative patterns in data [45]. The third term widely used in literature is that of *machine learning*. It evolved out of a number of fields, such as applied statistics and pattern recognition and arose out of a subfield of Artificial Intelligence (AI) [45]. Machine learning focuses on the usage of computational methods for improving performance by automating knowledge acquisition,

gained from experience [49]. The focus lies primarily on the automation of knowledge discovery to replace the inefficient human-centered data analysis [50]. This automation process is expected to increase its accuracy and/or efficiency with the discovery of regularities in *training data sets*[1] [50]. The theory behind providing machines the capability to learn from one situation and apply this knowledge on another is not new and can be traced back to the 1980's [51]. Since machine learning methods have been deployed widely, the related disciplines generated ever-closer ties, resulting in the separation between the fields being fuzzy [45]. Nevertheless, a distinction between data mining and machine learning will be provided at this point. Both disciplines aim to retrieve data patterns in order to create models, which in turn can be used to evaluate new data sets. Machine learning goes one step further by aiming to improve its models with each piece of new data [45].

Model building and usage Both data mining and machine learning employ the building of models using data mining methods and algorithms [45]. This is only one part of the utilization of data mining. A distinction has to be made between the *mining* of data itself (in order to find patterns or build models) and the *usage* of the results of data mining. Figure 2.3 illustrates this distinction. A data mining

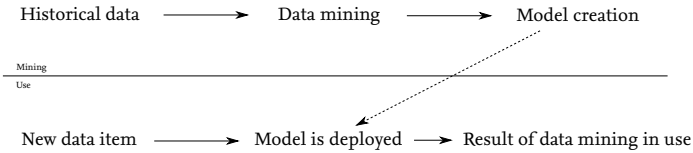


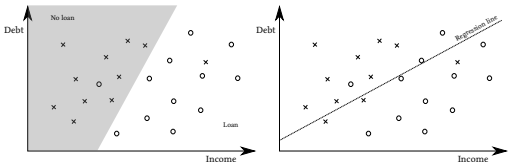
Figure 2.3.: The difference between data mining and usage of data mining results, after Provost and Fawcett [45].

algorithm or method is first applied on historical data. By doing so, models are created based on the patterns found in data. These models are then deployed for the evaluation of new data and return a result. One example is the creation of a probability estimation model using historical data. It is then applied to a new, unseen data item, and generates a probability estimate for it. [45]

2.2.3. Data mining and machine learning

As indicated in 2.2.2, the boundaries of data mining and machine learning often overlap. It is further important to note that statistics play a large role in machine

[1] Training data refers to data being used to train and/or establish a model. Its values need to be known in order for a correct calibration of the model taking place. [50]



(a) Linear classification denoting class *no loan*.

(b) Linear regression describing debt as a function of income.

Figure 2.4.: Comparison of classification and regression methods, after Fayyad et al. [48]. A data set is pictured, with crosses for data points without a loan and circles for ones receiving a loan. Illustrated in the left figure is a determination what the income/loan ratio must be to predict whether future data points fall into the *no loan* class. In the right figure, a linear regression is shown, in which debt is determined (or *fitted*) as a function of income.

learning, resulting in yet more overlaps between the fields [50]. The following sections provide first a distinction between classification and regression, after which an overview of machine learning algorithms is provided.

Classification and regression

A common method for developing predictive models is **Classification**. This methodology aims to predict for each data point, which of a number of classes this point belongs to [45]. New data points are effectively given labels, indicating to what group or entity these belong to or can be associated with [52]. Of similarity to classification is *probability estimation* [45]. Instead of the model classifying what label a data point is to receive, the model yields a score representing the probability of the data point belonging to each class. This similarity results in classification models usually being able to provide probabilities as well and vice versa [45].
Another method, **Regression**, aims to estimate or predict a numerical value of a variable for each individual data point [45]. This method determines the amount an individual can be associated with a variable. A simple example is the service usage of a telecommunications client; clearly, the client can be associated as being a user, with regression determining the actual amount of usage. Regression and classification are therefore related. Classification determines *whether* something will happen, while regression aims to gauge *how much* something will happen. [45, 50] These two methods represent the most common methods in pattern recognition problems [53]. Both methods and their differences are illustrated in figure 2.4.

Machine learning algorithms

This section focuses first on the generic schematic of any machine learning algorithm, after which an overview of a number of commonly-used algorithms are presented, classified into groups and their respective advantages, disadvantages and complexities discussed.

General machine learning schematic While every machine learning algorithm relies on a different strategy to build a predictive model, a single logical high-level structure is applicable to all, illustrated in fig. 2.5. Core to this structure is the

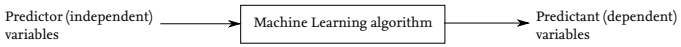


Figure 2.5.: High-level mechanism of every machine learning algorithm, with input/predictor and output/predictant variables.

machine learning algorithm, as illustrated in the center in fig. 2.5. Any input to any algorithm is called a *predictor*, also called an *independent variable*, since these are considered given variables. Output variables, on the other hand, are dubbed the *predictants* or *dependent* variables. Hence, in a mathematical notation, the goal of any machine learning algorithm is to estimate function $f()$ in equation 2.3:

$$Y = f(X) \tag{2.3}$$

Where Y resembles the prediction in the future, given a set of input variables or conditions X. Would the function $f()$ be known, the predictant could be calculated directly. Since nothing is known about this function, algorithms need to be employed to determine an approximation or ideally, a determination of $f()$. Every algorithm relies on one or more assumptions and effectively represents a model of possible variable correlations. Hence, the drawback of any model is that it is limited in the degrees of complexity and therefore may not accurately imitate the behavior of any physical phenomenon. [54] A prime example of such a function is a simple linear regression:

$$Y = w_0 + w_1 \cdot X \tag{2.4}$$

Linear regression assumes a linear relationship between predictors and predictants, as shown in eq. 2.4. The goal of the algorithm is to find the weights w_0 and w_1, which ensure a best fit between in- and output variables. [54]

Time series is another competency field focusing on forecast prediction. As the name says, these methods aim to capture the behavior of the variables sequentially over time. An example is the prediction of stock market trends. [55]

Overview of machine learning algorithms Machine learning has been chosen in this work, as the methods stand in stark contrast to the approach undertaken by meteorological entities. The latter rely on deterministic approaches. Instead in this work, the focus is on the question whether a purely statistical approach while omitting physical correlations, is worthwhile. Machine learning algorithms each have their own assumption on the underlying data distribution. As such, a multitude of different machine learning algorithms exist. Fig. 2.6 illustrates an overview of a number of algorithms. This list is far from a complete representation of all methods, but highlights the most commonly used algorithms. Also, this overview represents only one way of ordering the various algorithms. The focus herein lies in identifying ones that support supervised learning, hence yielding the depicted classification. For this work, the following algorithms are selected for evaluation

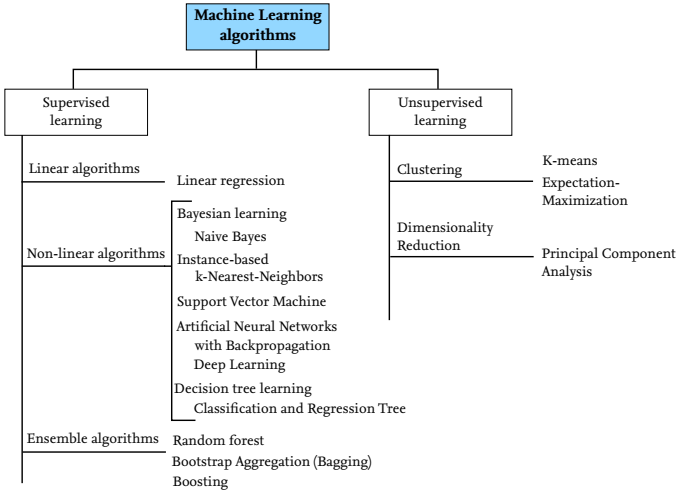


Figure 2.6.: An overview of machine learning algorithms. These are classified into algorithms that correspond to supervised and unsupervised methods.

of their respective predictive power: a linear regression; 5th, 10th and 15th degree polynomial regression; a support vector machine; a decision tree; a boosting and a k-Nearest-Neighbor algorithm. Each of these carry both advantages and disadvantages in the field of prediction performance. These are explained in greater detail in table 2.2, where each algorithm is graded by its general performance in a number of characteristics.

The distinction between supervised and unsupervised methods is explained by

Friedman et al. [56] as the difference in learning methodology: For supervised methods, the training data bears values for input and output data, allowing for a correction whilst learning. For unsupervised methods, the output data is unknown and data patterns need to be inferred from input data only, without the possibility of correcting errors.

Supervised learning algorithms

This section focuses on supervised learning algorithms. A further distinction can be drawn between algorithms which model linear and non-linear behavior. A third category, called *ensemble algorithms*, groups algorithms that work by combining the predictive power of a number of weaker models in order to generate one strong predictive model [56].

Linear and higher-degree regression Linear regression aims to fit a linear function to a set of data points, as illustrated in fig. 2.4. In principle, this method aims to determine the weights w_0 and w_1 in eq. 2.4, in order to generate the best fit. [54] This does not need to be limited to a linear regression, as function 2.4 can be extended to fit non-linear behavior by increasing the degree of x. [56] In this way, more dynamic behavior in data can be fit, possibly leading to better prediction accuracy. Higher-degree (or -order) polynomial fits can be performed by extension of the base X_j to $X_2 = X_1^2$, $X_3 = X_1^3$ or higher.

Instance-based methods k-Nearest-Neighbors (kNN), allocated in the class of instance-based methods, relies on distance as its main metric [56]. The algorithm also does not build a model, but rather determines the distance of a new data point to every available training data point. Fig. 2.7 illustrates this schematically for an exemplary $k = 6$ kNN setting. The k number of nearest points are then used to classify

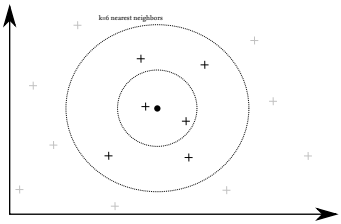


Figure 2.7.: An identification of $k = 6$ nearest neighbors in 2D space.

the new point by majority voting. For calculation of the distance, various distance

functions can be utilized, however commonly, Euclidean distance is used:

$$d_{(i)} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{2.5}$$

This equation 2.5 describes the distance between points q and p, in i dimensions. It is however necessary to normalize each dimension prior to calculating distances, as different dimensions might be measured in different units. [56] Normalization can be performed in numerous ways, with typically *feature scaling* (bringing all values into the range of $[0, 1]$ being widely utilized in machine learning:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.6}$$

With X' representing the normalized value, X the original value and X_{min} and X_{max} representing the minimum and maximum values in a range of values.

Support Vector Machines Support Vector Machines (SVM) were developed to create boundaries in feature spaces that are not linearly separable, i.e. in which classes overlap and in which a clear linear boundary cannot be drawn [57]. To do so, an input vector is transformed into a high dimensional space using non-linear mapping. In the resulting linear space, separation planes to define classification boundaries can then be inserted. Support vectors are first used to determine the optimal, i.e. maximum distance between the points. The optimal hyperplane is then placed at the middle distance of the vectors. [57] Support Vector Machine (SVM) commonly rely on kernel methods, which can be described as estimating the regression function of the local neighborhood of a given point. The latter is described by a kernel function $K(x, x')$, with a common kernel function being the Radial Basis Function (RBF) kernel, which is based on the Gaussian density function [58, 56]:

$$K(x, x') = \exp\left\{-\gamma \left\|x - x'\right\|^2\right\} \tag{2.7}$$

With $\|x - x'\|^2$ effectively representing the Euclidean distance of x from x' and γ being the variance of the Gaussian density, hence controlling the width of the neighborhood. [56] This kernel function is then inserted into the normal form for SVM classification [59]:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b \tag{2.8}$$

Decision tree learning Decision trees are a widely-utilized way for the approximation of discrete-value functions, with the learned function represented by a decision tree [54]. As learning progresses, the tree grows more complex. For human readability, decision trees are often represented as a set of *if-then* rules. Classification of instances are performed by the branches down the tree from the root to a leaf node. Fig. 2.8 illustrates this. Each node represents a test of an attribute of the

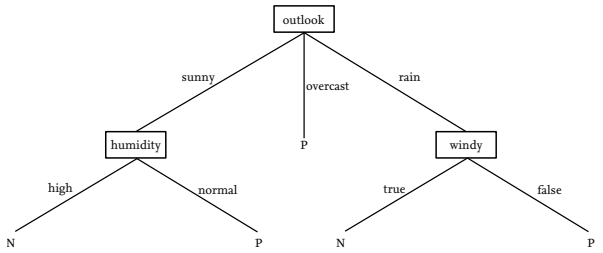


Figure 2.8.: An exemplary weather decision tree network, after Quinlan [60]. Illustrated are multiple branches, representing possible outcomes. At the very bottom, *Classes* represent the end of the branches, with *N* being negative and *P* positive.

relevant instance. Every node features branches, which represent possible values as outcome of the test. [54] Once a terminal branch is reached, a decision is made. In fig. 2.8, this is exemplified by a classification of weather into categories, negative and positive. Assuming that the outlook is sunny, the left branch is selected. The next test revolves around the humidity. If humid conditions are expected, then this weather combination is classified as negative, whereas normal humidity is classified as being positive. Effectively, decision trees can be defined as a hierarchical set of rules, against which an instance is progressively being evaluated. [50] While decision trees feature greatly in terms of construction speed, they lose out on accuracy. This is primarily due to the nature of how decision trees divide a space into branches. [56]
Computationally, a decision tree is created by recursively partitioning the available data. [61] The aim is that at each partition, the outcome of one leaf differs as much as possible to the other. By starting out at a root note, the best split is determined according to a pre-defined criterion, e.g. deviance or gini. The process is repeated until another criterion defining stoppage, is reached. Commonly, splits are determined by finding the one with the maximum information gain.

Random forests Random forests are essentially an extension of decision tree learning. Instead of creating a single tree, multiple trees are created and lastly, averaged.

Each tree is created based on a random vector, which determines the growth of every tree [62]. This random vector also determines the training values selected for building each tree. The idea behind this algorithm is that the errors generated by each of the trees continuously converge to a limit. Random forests follow the *Strong Law of Large Numbers*. This proven theorem states that as the number of trees increase, a convergence point will always be found. This assurance of convergence is additionally the reason why random forest cannot be overfit with a growing number of trees. [62]

Boosting Boosting aims to generate a single strong classifier or predictor from multiple weak ones. [56] This is realized by generating a model from the training data, upon which a second model is created, with which the errors from the first model are corrected. This process is repeated over and over until the training set can be perfectly predicted or until a certain number of maximum models are reached. Of the many different algorithms, *Adaboost* is by far the most commonly used [56]. It relies on building decision trees; after the first one is build, the tree's performance on every training data point is utilized to determine how much attention the following tree should allocate to every training point. Data points harder to predict are given more weight and ones easier to predict are given less weight. This process repeats, with every following tree updating weights, until the training set is predicted perfectly. [56]
By doing so, prediction accuracy of decision trees can oftentimes be greatly increased, which simple decision trees commonly lack. On the other hand, a number of advantages of decision trees, such as training speed and interpretability are lost. [56]

Comparison of algorithmic characteristics All algorithms are based on one or more assumptions on the distribution of data. Linear regression for example assumes that a linear relationship exists in the data. Fitting this type of regression to data that actually exhibits a linear relationship may generate good prediction results, however with data showing non-linear tendencies, the trained regression may perform poorly prediction-wise. The algorithms detailed in chapter 2.2.3 all feature different assumptions and will bear advantages and disadvantages in different characteristics. In most cases, performance of a given algorithm in an arbitrary situation or problem is unknown in advance to data analysis. [56] Some characteristics and four algorithms' performances in each of these are summarized in table 2.2.
Table 2.2 is read with the following legend: ● = *poor*, ● = *fair*, ● = *good*.
Summarized are the performances in the stated characteristics for SVM, trees, all types of regressions and kNN. In the course of a data analysis, it is important to test a number of different algorithms in order to identify those that best generate pre-

Characteristics	SVM	Trees	Regressions	kNN
Handling of various kinds of data inputs				
Handling of incomplete data sets				
Robustness to outliers				
Efficient scalability towards large N				
Handling of irrelevant input variables				
Extracting linear combinations of data features				
Interpretability of results				
Algorithmic prediction power				

Table 2.2.: A number of characteristics of various supervised learning algorithms, after Friedman et al. [56].

dictions. Table 2.2 is to be treated as a general reference, instead of a prediction of how these four algorithms will perform in the actual analysis in the work herein. Notable observable tendencies are that SVM excel when the extraction of linear combinations is concerned, whereas this type of algorithm performs poorly when different types of data are mixed (e.g. continuous and categorical variables), when data sets have missing values or when the data is outlier-prone. This can be explained by the core logic of an SVM, as it divides the space by establishing support vectors. These vectors are planes in d-dimensional space (d being the number of dimensions in the input data set), hence any instances missing data for a number of dimensions will result in a failure to build vectors in areas in which values are missing. Trees in general on the other hand, including *forests*, tend to be the other way around. As they divide the data space up recursively setting boundaries in all dimensions, they are not as susceptible to large volumes of data. This is due to trees not dividing by vectors, as SVMs do. By doing so, they can easily handle outliers, as these can collectively by divided to be part of a branch, separate from the main branch. The disadvantage of trees is their prediction power, if their complexity is not sufficient. By increasing a tree's complexity results in greater processing time needed and a decrease in interpretability of the results. Another algorithm exhibiting robustness to outliers is kNN. Since its working logic is to find similar past instances, its predictions are not influenced by the greater population of data points. In contrast, this is one fundamental disadvantage of regressions in general,

as these determine the best fit. Only the points in the vicinity of the fit will likely generate an accurate prediction using the regression function. Outliers which are especially far from the function will exhibit a poor prediction power. On the other hand, regressions carry the inherent benefit of being easily interpretable.
These characteristics serve as rough guidelines when designing a data analysis and may aid during the evaluation of which. Moreover, table 2.2 identifies areas of competencies which could help when deciding the types of algorithms to use.

2.2.4. Big Data applications in aviation

Applications of Big Data in aviation has been rather limited. Oftentimes, authors present their own definition of Big Data, as the term is still relatively young with no single definition as yet defined by the research community. Applications of Big Data technologies are listed in this section if the author publishes it as work in Big Data. In this work, the author's definition of Big Data is presented in 2.2.
In academia, a number of authors have recently published work on analysis on big data sets, with relevance falling to the passenger, maintenance and airspace realm. Akerkar [30] analyzed passenger data sets of 40 Gigabytes (GB) in size. The result included information on airline performance, market and passenger booking patterns. Big Data activity in maintenance can be found in the realms of the analysis of test data [63]. Further sources [64, 29] indicate research work in the wider Maintenance, Repair and Overhaul (MRO) area, such as by General Electric [65]. General focus of these activities is the prediction of the time of failure of parts. The general trend for airplane manufacturers over the past decade has been to install sensors on board airplanes in order to track up to 1000 variables during flight [29]. Such a wealth of variables and data sources yields a large amount of data generated per flight. As such, technologies able to handle these amounts of data need to be used to allow processing in a feasible timeframe.
Activity for Big Data applications on ATM issues can be found in [41, 42]. Boci et al. [41] focused on the analysis of large-scale Automatic Dependent Surveillance - Broadcast (ADS-B) using a data lake, while Ayhan et al. [42] worked on predictive analyses of airspace movements with an underlying data lake structure running with algorithms provided by IBM.
Big Data applications in the aviation industry remain significantly more advanced and far-flung. This may be due to Big Data being primarily a focus of companies hoping to gain an upper hand over their competition, by employing these methods. The focus of these applications can be grouped as solutions aiming to drive down costs for aircraft operators in fuel and the MRO realm. Notable companies involved in fuel efficiency solutions include Airbus and IBM [66], who are collabo-

rating to provide so-called *Smarter Fleet* solutions. These rely on Big Data tools and solutions. Other providers in the fuel efficiency realm are Lufthansa Systems [67] and General Electric [68], who both provide platforms for airline customers for various efficiency solutions.
The second cluster of Big Data applications in the MRO realm is lead by widely-known engine manufacturers, such as Rolls-Royce [69], General Electric [68] and Pratt & Whitney [70]. All companies aim to provide a system with which predictive analytics based on large-scale data collection can be provided to the airline customer. Aircraft manufacturers Airbus [66] and Boeing [71] have also commenced development in this realm, with the goal being to provide an increase in operational performance through data analysis of vast amounts of data collected while inflight.

2.3. Weather forecasting and forecast verification

Forecasts in almost all instances carry the belief that even inaccurate or limited knowledge on a matter is in all cases better than having none [72]. This is also true of weather forecasts, among many others disciplines [72]. Weather forecasting in the U.S. and western Europe commenced in the years between 1850-1870, in parallel to the establishing of regional and national weather services [73].
In the early days of weather forecasting, predictions were based on subjective opinions [74]. Over time, the necessity to utilize objective methods has gained more momentum. Statistical methods, particularly regression, as well as Monte Carlo methods have been implemented. [74] The idea of using numerical methods and physical properties to determine forecasts was developed shortly after, yet proved infeasible without the necessary machines to conduct vast calculations [75]. With the advancement of computers and increasing computational power, these methods could then be applied. A more advanced understanding of the physics of the atmosphere as well as more data sources also added to the increased use of numerical methods. [75] These are used, under implementation of Atmospheric Circulation Models (ACMs), to create forecasts of wide geographic areas [2]. These grids are rather coarsely-grained with a resolution of approximately 50-100 kms. According to Cofiño et al. [2], a number of meteorological phenomena including rainfall, quantitatively vary on smaller scales. This effectively leads to ACMs not providing a detailed forecast of local scales for these phenomena. This shortfall has lead to the application of statistical and machine learning techniques to weather forecasting. These methods rely on databases containing historical weather observations

to train models, which are then used to predict future meteorological phenomena. [2] These probabilistic methods differentiate themselves by avoiding the use of deterministic models based on physical properties [76]

2.3.1. State-of-the-art of weather forecasting

This section begins by stating works on numerical methods used for weather forecasting. This is followed by a number of recent machine learning applications, in order to provide the reader with the current state-of-the-art regarding the proliferation of probabilistic methods. These will include work based on regression approaches, hidden Markov models and neural and Bayesian networks. A short paragraph will also give a short description of combination approaches.

The majority of work in the realm of weather forecasting rely on generative approaches, in which the weather is simulated by numerical methods [77, 78]. As these do not yield detailed descriptions of local weather phenomena, statistical and machine learning techniques have started to be applied in this field. Regression and classification models have been employed [79], which aim to decrease the uncertainty of regional weather forecasts. Another approach [80] has been to utilize hidden Markov chains, which assume unobserved weather patterns and states and which follow a Markov chain. Neural networks have too been applied in weather forecasting [81, 82, 83], however limited, due to a number of fundamental difficulties encountered when applied in the realm of meteorology as compared to regression methods [82]. Work has also been performed to connect ACMs with predictive models. The patterns predicted by the models are fed into observational databases [84]. By doing so, sub-grid details can be calculated. This process is widely known as *downscaling*. However, drawbacks exist when applying this model, as statistical independence between different variables is assumed and important information thereby ignored [2].

Although machine learning methods have been successful in a number of fields and tasks, applications in the weather forecasting domain has been limited. Bayesian methods are the only major exception and work has been done on the prediction of precipitation and general weather one day in advance [2, 85]. The major benefit exploited when applying Bayesian networks is the fact that these can be ideally implemented to discover dependencies among variables [2].

More recently, wind prediction using publicly-available flight tracking data has been presented [86]. By using a Bayesian framework with Gaussian Processes, the feasibility of leveraging aircraft as an inflight sensor network for wind forecasts could be shown. Another work by Grover et al. [87] focuses on predicting weather while taking the tightly coupled weather variables into account.

2.3.2. Forecast verification

Tendencies to question and determine the quality of forecasts started shortly after the establishment of national weather services, with Finley's work on tornado forecast verification in the U.S. [88] being one of the first works in this field. This publication accelerated the drive to verify forecasts in the following years [89, 74]. Verification of the forecasts on how accurate these predicted the real weather has been of great importance and continues to do so [72]. A primary challenge for meteorologists has been the establishment of a scale for *goodness* for weather forecasts. While a number of such scales have been proposed, agreement has not been reached on what the most useful is [90]. One of the most well-known researchers in the field of meteorology, Allen H. Murphy presented a definition of a good forecast [91]. The definition is threefold, of which two are very familiar and widely accepted by forecasters. One of which describes a forecast as good, if the forecast conditions do not deviate significantly from the observed conditions. [91] A general framework for forecast verification was presented by Murphy and Winkler [92], while a further publication approached the issues generated by forecast verification as in complexity and dimensionality [93].

Verification methods were surveyed in the late 1980's by the World Meteorological Organization (WMO) [94]. Currently, the WMO publishes recommendations on verification strategies and scores, in order to support exchanges of verification scores between different locations [72][2]. Member states of the European Union provide annual reports on the verification of forecasts in their respective national weather services. At national level, forecast verification strategies may vary. This may be due to the varying forecast purpose of each service's user group. A broad overview of current verification methods is provided by the *Joint Working Group on Forecast Verification Research*[3]. [72] While national weather services mostly verify their forecasts, this may not be standard practice for private forecasting companies [72]. Studies commissioned by the Royal Meteorological Society have shown that large discrepancies in the quality of meteorological forecasts exist, with some users seemingly being indifferent to forecast accuracy [95, 96]. The authors also criticize the lack of uncertainty estimation of forecast results.

2.3.3. Meteorological outputs

A number of different outputs are produced by meteorological entities, such as the National Weather Service (NWS). Each represents a different use case and is therefore generated in a different manner. The Research Data Archive of the

[2] http://www.bom.gov.au/wmo/lrfvs/users.shtml
[3] http://www.cawcr.gov.au/projects/verification/

National Centers for Environmental Prediction (NCEP) retains a number of data set types, of which three relevant ones are described in the following:

- **Forecasts:** Forecasts are available on a wide range of variables and are generated in different time steps. The type used in this work is generated by the GFS four times a day, at 00z, 06z, 12z and 18z. These have a forecast range of up to 36 hours in advance with, among others, a spatial resolution of 0.5°across the globe. A GFS numerical forecast run is initialized using the analysis generated six hours prior. Due to the time needed to perform a Numerical weather prediction (NWP), a forecast needs to be initialized early enough so that it is available at the mentioned times of day. [97, 98]

- **Analyses:** Analyses are a snapshot in time and are produced by a large number of observations on an irregular grid, with the goal of producing a depiction of the state of the atmosphere at a point in time. This is in contrast to forecasts, which describe a condition in the future. An analysis normally incorporates around 10% more observations than a forecast. In contrast to forecast generation, the initialization of an analysis will therefore have to wait until a sufficient number of observations have arrived to start the process. An analysis for the state of the atmosphere at e.g. 00z will likely only be available an hour or so later, yet in time for the initialization of the next forecast run. [98, 99]

- **Re-analysis:** Re-analyses are analysis runs on a fixed atmospheric model and software system. They are run across large numbers of years, as in the case of the 31-year re-analysis run (1979 to 2009) [100]. Re-analyses use only a single model, as well as one data assimilation system. The benefit of this is that the re-analyses are not affected by a change in method, as can be the case with analyses [101]. For the re-analysis, all available data, from e.g. radiosondes, aircraft and Aircraft Communications Addressing and Reporting System (ACARS) data, surface observations, ocean surface wind speeds and satellite wind observations among others, are included [100]. Therefore, this kind of data is considered the best estimate of the state of the atmosphere. [100, 102]

Fig. 2.9 illustrates these three types of meteorological outputs. At at arbitrary time $T1$, an analysis and a forecast are generated. The analysis depicts the meteorological conditions at $T1$, while the forecast pictures conditions at $T2$. At a later point in time $T3$, a re-analysis is generated which describes meteorological conditions valid at every past time stamp, including $T1$ and $T2$.

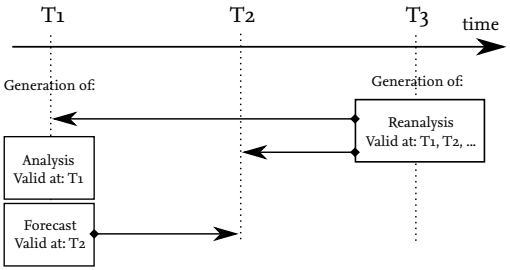


Figure 2.9.: Differentiation of analyses, forecasts and reanalyses, their dates of creation/generation and validity.

2.3.4. Data variables and units

Meteorological entities commonly provide wind speeds in a two-component fashion, *U wind component (UGRD)* and *V wind component (VGRD)*. The former is positively defined in a northward and the latter defined in an eastward direction. Wind speeds are expressed in units of meter per second $\frac{m}{s}$. [103]
Another two variables, which are also provided to common flight planning tools as input data, are *Geopotential height (HGT)* and *Temperature (TMP)*. The unit for TMP is Kelvin *K*, while HGT is expressed in gpm. HGT is of particular importance, as the values of this variable are used to compute the actual geometric height or altitude. From the standpoint of a meteorological organization, height is commonly declared along a geopotential. Measurements are often conducted using radiosondes, the geometric elevation of which is dependent on pressure, temperature and the gravitational acceleration at any arbitrary point [104]. Due to constant variations in the atmospheric properties, it is therefore standing to reason that the general variable for altitude not be one expressed by a geometric, but geopotential. This stands in contrast to the usage of barometric altitudes by aircraft. The benefit of using geopotential in meteorology lies in the geopotential only being dependent on the variable gravitational acceleration along its geometric altitude. The common unit for this purpose is the *Geopotential meter (gpm)*, its definition expressed by Wright [104] as:
"The definition of geopotential is the potential energy due to gravity of a unit mass of air at some point above a standard position (i.e. zero energy), usually mean sea-level, and is measured in a positive sense vertically upward."

gpm as a unit itself is also defined by the same author as in equation 2.9.

$$h_n = \frac{1}{9.80665} \int_0^n g(z)dz \tag{2.9}$$

With n the index for the number of the layer, the constant 9.80665 representing the standard acceleration of gravity and $g(z)$ being the respective gravitational acceleration at a geometric height z. The equation for conversion from geopotential to geometric altitude is also expressed by [104] as in equation 2.10.

$$\begin{gathered} z_n = \frac{h_n R_e}{Gr - h_n}, \quad Gr = \frac{g_N R_e}{9.80665} \\ g_N = 9.80616 \left[1 - 0.002637 \left(cos(2N)\right) + 0.0000059 \left(cos^2(2N)\right)\right] \end{gathered} \tag{2.10}$$

With R_e the radius of the earth at latitude N, g_N the equation for the gravity at N and Gr being the gravity ratio. The earth radius $R_e(N)$ at latitude N is defined by equation 2.11.

$$\begin{gathered} R_e = \frac{a\sqrt{(1-e^2)^2 \sin^2(N) + \cos^2(N)}}{\sqrt{1 - e^2 \sin^2(N)}}, \\ e = \sqrt{1 - \frac{b^2}{a^2}} \end{gathered} \tag{2.11}$$

Calculation of the local radius at any latitude is necessary due to the earth's eccentricity e. Furthermore, two constants are needed for the calculation: the equatorial radius of the earth at $a = 6378137.0m$ and the polar radius at $b = 6356752.3142m$ [105].[4]

2.3.5. State of the art of weather uncertainty handling in aviation

Uncertainties in weather phenomena have always represented a research effort throughout history, as described in 2.3.2. The aviation industry has not been exempt from this. A number of research groups, namely the Europe/USA Air Traffic Management Research and Development Seminars, ComplexWorld[5] and SESAR's *Innovation Days*, have recently focused on the handling as well as impact of weather uncertainty to aviation.

[4] These two radii are also known as the *semi-major axis* and *semi-minor axis*, respectively. [105]

[5] This research network represents the research themes of the long-term objectives of SESAR and initiated by research body Innaxis.

The mentioned groups all lay a strong focus on an ATM viewpoint. Prior to the launch of ComplexWorld, research had already begun on examining the air traffic flow under impact of weather uncertainty [106]. In this work, a number of algorithms for weather avoidance are presented, which when simulated, yielded a significant reduction in overall ATM delay.

Following work involving the uncertainty of weather can be found under the initialization of the general *Impact modelling of adverse weather on ATM performance* program by ComplexWorld. Two tasks have been primarily defined in this program: quantification of the uncertainty of specific weather phenomena (thunderstorms, precipitation) as well as a focus on the avoidance of adverse weather by aircraft. For the first focal point, Sauer et al. [107] is the most recent work. The authors describe a methodology for the uncertainty analysis of thunderstorm nowcasts. This topic is the result of a prior research focus on adverse weather avoidance modelling. Work has primarily involved the so-called Adverse weather diversion model (DIVMET) [108]. This model has been deployed for the optimization of flight route changes, among other applications [109, 110, 111]. Other work, by Schilke and Hecker [112], proposed an on-board system, with which the flight route of the aircraft is altered around convective weather. The aim is to optimize the route as early as possible, so as to fly the most efficient one possible.

In parallel to this research, Tino approached the topic of wind velocity uncertainties [113, 114]. In [113], a methodology is presented for estimating wind errors using information acquired through ACARS messages. More recently, the author published work on wind models and stochastic programming algorithms [114]. These are applied on several high-density routes in the continental U.S., yielding averaged wind uncertainties along these. The aim of this work is to improve the air traffic network's predictability of traffic streams.

Uncertainties in forecasts have long been considered to be mainly the result of uncertainties in numerical simulations' initial conditions [115]. To achieve a greater likelihood of capturing the correct initial conditions, the Ensemble Prediction System (EPS) by the European Centre for Medium-Range Weather Forecasts (ECMWF) was set up in 1992. This system utilizes a set of different initial conditions for simulation, yielding different forecast outcomes. Buizza et al. [115] investigated a methodology to increase the spread of the ensemble while improving the forecasting accuracy for some parameters. This methodology was later integrated into the operational EPS in 1998. [115]

Further research efforts in the context of EPS have been conducted by Candille et al. [116]. The authors utilized observational data to verify the EPS at the Canadian Meteorological Centre (CMC). Verification was performed against the observed data, culminating in a revision of the working EPS. Usage of such a sys-

tem in a pure aviation setting has recently been a major focus of both ComplexWorld and SESAR's *Innovation Days*. Cheung et al. [3], in the course of the *IMET*[6] project, evaluated the sensitivity of flight durations due to uncertainties in numerical weather predictions. Each ensemble member forecast of an EPS was used to predict one trajectory, yielding a range of different trajectories due to each member's simulated wind predictions. All ensemble members are by definition equally probable, resulting in all predicted trajectories also being equally probable. In a following publication [4], this approach was further deepened by the calculation of each trajectory's contingency fuel. By calculating the amount for each ensemble member, a Probability Density Function (PDF) of the predicted contingency fuel could be determined. This effectively resulted in an estimate on the uncertainty of contingency fuel due to weather uncertainty. In the case of a PDF with a low Standard Deviation (SD), the uncertainty of the required contingency fuel is low and high in the case of a high SD. This knowledge is assumed to support airline decision making, as the process sheds light on a probable interval of the amount of contingency fuel. [4]

It is important to note that the application of EPS on flight planning provides a means of quantifying the spread of needed contingency fuel due to uncertainty. The most probable forecast itself is not much improved with the ensemble approach [117]. This challenge is the focus of the work herein, for which a proposed application is presented in the following section.

2.4. Proposal for a machine learning application in flight planning

Solving the optimization problem in flight planning of finding the most cost-efficient route has heralded the development of two solution approaches to date, as described in 2.1.4. Departing from a 2D and embracing a 4D network solution approach, the aim has been to increase the computational efficiency of the flight planning logic. While the change in approach has the potential to avoid a deterioration of the quality of the solution [10], it aims to increase the quality of flight planning through efficiency. In this thesis, a different and novel approach towards the improvement of the flight planning process is proposed. Instead of aiming to optimize the efficiency, the proposed approach targets the effectiveness of the process, by improving the quality of the input data. This approach is in theory not contradictory to the type of network solution approach. The two approaches may

[6] IMET: Investigation of the optimal approach for future trajectory prediction systems to use METeteorological uncertainty information.

even complement each other as both target an optimization in different areas of the problem and by different means.

2.4.1. Proposal outline

This section presents first an assessment of foregone work in the field of uncertainty handling in an aviation context. Work on uncertainty handling on a wider context is presented in 2.3.5. In this section, only specific research work which share similarities and overlaps in methodology are presented. By evaluating these prior works, a research gap is then identified, for which a solution for closing it is in turn proposed.

Prior work

The proposed idea envisions a method for the quantification of the uncertainty in wind forecasts used in flight planning. As pointed out in 2.1.1, uncertainty is not considered in most flight planning tools and the contingency fuel is carried along to account for deviations along the flight path due to meteorological conditions. Efforts to quantify and better handle the impact of weather uncertainty in aviation have centered largely on decision support while in flight, such as [111, 112]. These proposals can not be directly applied to flight planning. Other efforts, such as by Kapoor et al. [86] have shown that a comparison between wind forecasts and inflight wind measurements can be successfully utilized for training a predictive model for wind forecast errors. The authors have also suggested that by applying their predictive model to flight planning, flying times as well as fuel usage could be minimized.

Leveraging inflight wind measurements has two major drawbacks; coverage and statistical relevance. Airspace above the U.S. and Europe feature a large number of aircraft, with 15.9 and 9.5 million controlled Instrument Flight Rules (IFR) flights alone in 2010 [118]. These numbers do not hold true for the majority of the remaining global airspace. While traffic is not non-existent, it may be sparser. The smaller number of flights means that a desired statistical relevance may not be achievable, potentially decreasing the confidence of the predictor. As flights commonly (in non-oceanic airspace) fly along airways, the coverage of such a system is limited to these paths. When considering flight planning, any waypoint may be taken into account when planning the most economical route. The effectiveness of a predictor as proposed in [86], may only be limited to common airways' waypoints.

Similar research has been done by Schwartz et al. [119]. In their research, the authors utilized data from Rapid Update Cycle (RUC) runs, as well as ACARS data. The goal of their research was to determine the accuracy of the forecast system. Average wind speed errors were determined in the process. The authors stopped

short of creating a system that could translate their learned insights into wind speed errors to predict future wind speed uncertainties. By leveraging exactly this prior work, Lee et al. [120] created a method that not only generated estimates on the wind uncertainty, but also determined the effect these uncertainties had when considered in conjunction with aircraft trajectory. A key characteristic of their method is the ability to correlate specific weather phenomena to variations in uncertainty. The authors utilize multiple forecast sets to create a forecast ensemble. Wind speeds and errors are considered for single point locations over the duration of the ensemble. In this way the goal of quantifying wind uncertainty by region, altitude and ensemble lead time can be determined. Underlying data for this research only comprises six successive RUC runs and only for the continental U.S.. It is also noteworthy that when calculating the impact of wind uncertainty on flight trajectories, the authors define a one sigma/standard deviation boundary for wind speeds with which the route deviation is determined. What is excluded from that work is the determination of the expected uncertainty by the quantitative nature of an arbitrary forecast.

Further development using RUC was pioneered by Zheng and Zhao [121]. With using each following RUC run's wind analysis as reference, i.e. the "true" value of the wind speed, the forecasts' wind errors was determined. While the research in this thesis aims to quantify wind speed errors by calculating the difference, a number of discrepancies need to be outlined. The data volume considered in [121] is limited and the authors themselves call for a larger data set to be examined to solidify their findings. Another stark difference lies in the type of data. While in [121] RUC forecast and analysis data is used, the research herein utilizes GFS forecasts as well as Climate Forecast System Re-analysis (CFSR) re-analysis data. The reason being that GFS forecasts are commonly used in flight planning and re-analysis data being the most accurate depiction of the state of the atmosphere [100]. Another differentiating factor lies in the handling of the uncertainty. The authors in [121] focus on quantifying the wind speed errors along regional and temporal correlations and present distributions of speeds. This represents one of the major differences, as one goal of the research herein is to not only develop the capability of providing a likely wind speed error in general for an arbitrary location, but predicting the varying error for quantitatively different forecasts. The reasoning behind this approach is that it is far from certain that the same error as well as its distribution will always be present in the same magnitude in all forecasts for any location.

Kahl and Samson [122] investigated the uncertainty in trajectory calculations in boundary layers, that resulted from low-resolution meteorological data. One of their results was that the interpolation used with this data was insufficient and

prone to yielding imprecise trajectories.
A number of authors investigated the impact of trajectory uncertainty from an ATM perspective. Mondoloni [123] addressed the multiple sources of forecast error and their impact on flight trajectories. Nilim et al. [106] proposed an algorithm for the risk evaluation of tactical route planning around convective weather. This dynamic routing strategy aimed to minimize the expected delay. Knowledge of this could help ATM improve its predictability.
The Europe/USA Air Traffic Management Research and Development Seminar also incorporated a drive to quantify the uncertainty of wind predictions and the resulting impact on trajectory accuracy. Papers from three consecutive seminars address this issue. Cole et al. [6] sought to improve RUC forecast runs by incorporating real-time aircraft observations and in this way reduce wind speed errors. On the other hand, Pepper et al. [124] proposed a method for accounting of uncertain weather information while utilizing Bayesian decision networks. The third kind of investigation by Clarke et al. [125] sought to determine the stochastic capacity of an airspace under weather uncertainty. An algorithm was proposed, with which the number of aircraft to be send into an airspace was determined. Routing guidance in the presence of uncertain events was also taken into account.

2.4.2. Research gap

From the summary of foregone work (see section 2.4.1), a number of insights along with three main prior work can be identified. All three research projects share certain aspects with this work. These are summarized in table 2.3. The first aspect is the backbone of the motivation of the work herein, as the goal is to improve the accuracy of weather forecast. While the last one closes the circle with an application of these improved forecasts onto the flight planning process, in order to estimate the value added. The novelty of this work is a fundamentally different approach to estimate forecast uncertainty, hence the aspect of machine learning. The remaining two aspects, statistical relevance and large geographical coverage are requirements to machine learning and flight planning respectively. Determination of wind speed uncertainty itself is not novel and has been pursued by a number of researchers. Both Kapoor et al. [86] and Zheng et al. [121], among others, focus on determining the uncertainty between data sets of forecast and actual wind data. Due to the very limited data size used, the authors themselves (such as Zheng et al.), call for an evaluation to *"...obtain more definitive conclusions, significantly more wind error data may be needed."* [121].
Their and other authors' [122, 123, 6] research focused on limited geographical areas, however showed the basic feasibility of efforts to determine wind forecast er-

	Zheng et al. [121]	Kapoor et al. [86]	Cheung et al. [4]	Cabos
Wind speed uncertainty determination	✓	✓	-	✓
Statistical relevance	-	-	-	✓
Large geographical coverage	-	✓	✓	✓
Machine learning on Big Data	(✓)	(✓)	-	✓
Application to flight planning	-	-	✓	✓

Table 2.3.: Comparison of prior work with the research gap being identified, which is to be covered by the work herein.

rors with historical data.
Kapoor et al., but especially Cheung et al. [4] pursued greater geographical coverage. The latter's work even allowed further application to flight planning along routes across the northern Atlantic.
On the aspect of machine learning on Big Data, the first two prior works in table 2.3 exhibit coverage, albeit only in part. While both pursue an effort to perform machine learning, they do so on data sets of limited size. On top of this, both restrict their activity to a single type of learning model.
An evaluation of the impact of determined wind uncertainties has only been performed by Cheung et al.. In the course of the IMET[7] project, the effect of forecast ensembles on the spread of fuel predictions by a flight planning engine was investigated.

This thesis aims to close the above identified gaps in research by proposing a method which covers all five aspects identified in table 2.3. The focus lies on the prediction of uncertainties in wind forecasts using underlying machine learning algorithms, trained on global historical wind forecast and re-analysis data. In order to achieve high statistical relevance, a prime focus of the method proposed herein is to rely on large data sets spanning years' worth of data. The data itself is envisioned to cover a large area from western Europe to South East Asia and across multiple altitude levels.

[7] IMET: Investigation of the optimal approach for future trajectory prediction systems to use METeteorological uncertainty information.

The prediction performance of the trained algorithms is first determined. In a last step, the impact of this prediction performance on the process of flight planning is evaluated. As such, an arbitrary prediction's effect on the accuracy of generated flight plans is to be gauged. In turn, this serves as a metric with which the feasibility of the herein proposed concept can be evaluated in the scope of an airline process.

2.5. Summary

This chapter provides an overview across the topics of flight planning, Big Data and machine learning and weather forecasting. Outlined are the fundamentals of flight planning, in particular the factors influencing different costs. On Big Data technologies, commonly employed data cluster structure and frameworks are presented. State of the art weather forecasting methods and data sources are elaborated upon, after which a motivation on weather uncertainty in the context of aviation is presented. A last section presents prior research works, before identifying a research gap, for which this thesis proposes a system.

3 Conception of a machine learning system for uncertainty prediction

This chapter describes the conception of the proposed machine learning application for the prediction of wind speed uncertainties in weather forecasts. First, a problem statement is presented, which is derived from the research gap identified in section 2.4.2. Serving as a motivation for the realization of a system that closes the research gap, a number of general requirements are then defined. Following these, a system architecture is presented, of which the main functions of the concept are described on a high level. The focus is laid on the overall architecture, connections and input and outputs of the different parts of this system.

3.1. Problem statement

The flight planning process relies to a significant extend on the wind and temperature forecasts, as detailed already in 2.1. Forecasts are assumed to be true in most flight planning engines [10]. No computational effort is invested in flight planning that deals with the uncertainty in forecasts and the impact that falsely or inaccurate forecasts have on the quality or accuracy of flight plans. This is particularly relevant in light of the highly dynamic nature of weather.

The current strategy to account for unpredictability of weather phenomena is to carry an extra amount of fuel. The obvious drawback to this methodology is that potentially unneeded fuel weight is carried. From a financial perspective, every unit of weight equals higher costs, as this weight needs to be transported from origin to destination. Thus, the question derived from this problem is whether it is possible to quantify the uncertainty posed in an arbitrary forecast (with which a flight is planned) and after accounting for this uncertainty, increase the predictability of the resulting flight plan. This issue of weather uncertainty in aviation is currently accounted for among others, by avoidance of adverse weather while in flight. However, at the time of writing, no solution with which to quantify the uncertainty of forecasts while in the phase of flight planning exists.

The goal of the methodology proposed in this thesis is to cover this gap. Envisioned is a method which predicts an expected uncertainty in an arbitrary forecast and thereby reduces the discrepancy between it and the actual condition.

3.2. General requirements considerations

Based on the problem statement worded in 3.1, the requirements for such a proposed methodology to be applied in a flight planning context can be defined. These are modeled after the input data needs in flight planning.

- **Intercontinental coverage:** Flight planning solutions are able to plan a flight between any two locations and can theoretically select any waypoint to be part of the plan. It is necessary that the proposed method covers a feasibly large area, allowing long-haul flights. Data sources must provide full coverage across these region. Should data not be available for a period of time on an arbitrary location, the concept will need to be designed with enough resilience to avoid a breakdown of the methodology.

- **Altitude level coverage:** When evaluating the uncertainty, a number of altitudes must be considered. Since flights may cruise on a number of altitudes and even change these in the course of a flight, a broad coverage is necessary to achieve a greater altitudinal resolution.

- **Evaluation of forecast patterns:** Flight planning solutions work with wind forecasts that are generated by the Global Forecast System (GFS). Typical forecasts feature a range of up to 36 hours with timesteps of either 3 or 6 hours. [97] It is necessary that wind speed deviations for an arbitrary point are predicted along the pattern comprising the values at each timestep. This pattern is effectively a time-series. The envisioned algorithm will need to be designed to perform a pattern recognition and return one or more predictions on likely deviations.

- **Statistical relevance:** For the method to provide accurate predictions and to maximize the confidence of these, a large data backlog will be needed. A timeframe of close to 10 years' worth of wind forecast and re-analysis data is to be used for the development of the method.

Core to this method are machine learning algorithms. The data collected (as mentioned in the last point in the above list) is used to train these, the idea being that an ever-increasing amount of data will further improve the algorithms' accuracies. A data cluster is be needed and utilized for the quantities of data to be efficiently

stored, handled and processed. The particular bottleneck is the computing power for data processing. A single machine may in theory be able to perform all necessary computations, in an unfeasibly large amount of time. Because of this limitation a Big Data cluster is required (see section 2.2 for more details), connecting multiple machines and being able to yield parallel processing capabilities aimed at shortening processing duration.

3.3. Data needs

Based on the general requirements laid out in 3.2, the following data needs can be derived:

3.3.1. Necessary data variables

With the nature of this concept being the prediction of wind speed uncertainties in wind forecasts, the primary data variable required is wind speed. As described in section 2.3.4, U wind component (UGRD) and V wind component (VGRD) are the two variables used to define wind direction and speed. Also needed in this thesis are Temperature (TMP) and Geopotential height (HGT) data. For the latter, the conversion from geopotential to geometric height is possible by application of equations 2.10 and 2.11.

3.3.2. Grid

Weather data provided by the World Meteorological Organization (WMO) or the United States (U.S.)' United States National Weather Service (NWS) is commonly distributed in General Regularly-distributed Information in Binary form (GRIB) format [126], in either edition 1 or 2. Data is highly compressed in this format, hence easing data transfer. Compression carries a drawback, as the data first needs to be converted to a different format prior to any data analysis. In this thesis, GRIB data was first converted to Comma-separated Values (CSV) format.
As the format's name implies, the data in a GRIB file is gridded. This makes particular sense for meteorological data, as any data value needs to be referenced by not only a temporal, but also a spatial tag. The spatial tag is described by latitude and longitude. Since one requirement (see 3.2) is that the concept works globally, all coordinate locations in a $0.5°$ by $0.5°$ resolution need to be taken into account. Fig. 3.1 illustrates this resolution and the remotest points considered in this thesis. The resolution by half a degree is chosen, as it is the resolution also serving as input to Jeppesen's flight planning software *FlitePlan Core*[1] [127]. A hypothetical

[1]This software will be used to evaluate the method described herein.

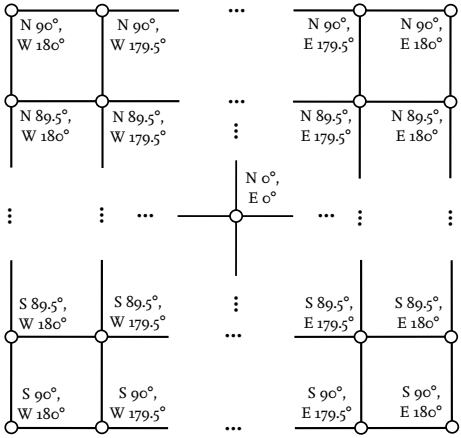


Figure 3.1.: Coverage of the type of grid utilized in this thesis. Shown is the spatial resolution of 0.5°, the four remotest corner points and one at $N0^\circ, E0^\circ$, indicating the grid's global coverage.

maximum of 720 longitudinal (W180° to E180°) and 361 latitudinal (N90° to S90°) coordinate points are considered, thus resulting in 259920 coordinate pairs. As is described in section 6.1.3, processing is limited to a fraction of these points.

3.3.3. Altitudinal levels

Due to the variability of the atmosphere, fixed geometric altitudes are not feasible. Instead, the altitude of a measurement is expressed as being valid on an *isobaric surface level*, the common unit used being *Millibars (mbar)*. To provide a thorough altitudinal coverage, in total, 14 pressure levels from 800 mbar to 150 mbar in steps of 50 mbar were selected. Fig. 3.2 illustrates such an altitudinal column for an arbitrary point. These two bounding values describe an altitude range that is commonly utilized by Instrument Flight Rules (IFR) flights and thus applicable to the flight planning process. Supporting this assessment is International Civil Aviation Organization (ICAO) *Doc 7488/3* [128], in which the International Standard Atmosphere (ISA) is described. Conversion tables in this document yield values for pressure level 150 mbar in the ISA equating to approximately $13,600m$ or $44,619ft$ and one of 800 mbar to around $1,900m$ or $6,233ft$.
With these pressure levels, an altitudinal grid is effectively produced, with a single point bearing two values for wind speed, one for temperature and another for geopotential height. The latter value determines the geometric altitude on which

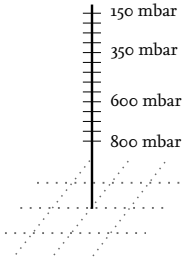


Figure 3.2.: Vertical pressure levels above one arbitrary 2D geographical point. Illustrated is the lowest level at 800 mbar and the highest at 150 mbar. Two intermediate levels are also shown, indicating that levels are defined by increments of 50 mbar.

all the prior values are valid. By adding a third value, altitude, to a point (defined by coordinates), a 3-dimensional (3D) point and thus a precise location can be defined.

3.3.4. Temporal considerations and forecast steps

In 3.2, the requirement for a large data set comprising multiple years to ensure statistical relevance is defined. The time span ultimately taken into consideration in this thesis ranges from November 01, 2006 to including June 30, 2016; representing 9 years and 8 months.

Forecast steps chosen range from 6 to 24 hr forecast steps, with a 6-hourly temporal resolution. Additionally, an *analysis* step is to be added to the data set. This step describes the meteorological conditions in the same way as forecasts do, with the difference that it is valid at the same time as it is created. Analyses are delayed from GFS forecast generation steps, so as to receive more observational data as input for the simulation run. The extra data available to analyses amounts to around 10% of the data volume of GFS forecasts. It is therefore a description of "current" conditions. [98, 99]

Analysis and forecast steps created together at the same date of creation form one *set*. During the above-mentioned time span, one set is obtained for every time step, i.e. every 6 hours. Fig. 3.3 illustrates how these sets are composed and connected to time stamps. 24 hr forecast data is the furthest forecast step used in this thesis, with even further steps unnecessary. This is due to maximum time span needed in the flight planning process. Typically, a flight plan is generated a few hours prior to takeoff, earliest at four hours and latest at 30 minutes prior to Estimated time of departure (ETD) [129]. Assuming an average lead time of 2 to 3 hours prior to takeoff and knowing that the longest civilian flight to have taken place until the

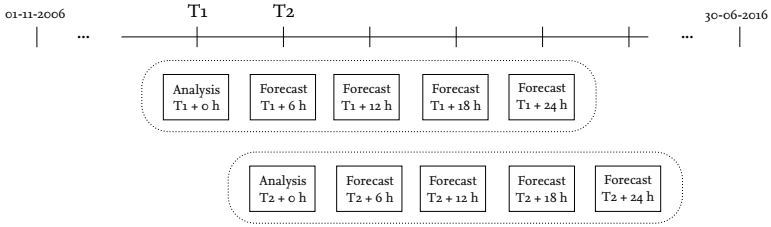


Figure 3.3.: Forecast sets and their composition. One set each is defined for arbitrary time stamps $T1$ and $T2$ in between the temporal boundaries of 01-11-2006 and 30-06-2016. The temporal difference between $T1$ and $T2$ is 6 hours.

time of writing was from Singapore Changi International Airport (SIN) to Newark Liberty International Airport (EWR) [130] with a flight time of under 18 hours [131], the maximum time span needed to be covered by forecasts is equal to around 21 hours. A forecast 24 hours ahead is sufficient and no further forecast steps are needed for single flight plans.

3.3.5. Data sets

As the proposed concept herein aims to quantify the uncertainties in wind forecasts, two types of data sets need to be acquired. The first one is a set of historical wind forecasts, with their deviation from the actually determined meteorological values to be evaluated. To do so, a second data set with records of the *true* or *actual* data is needed. While in theory, the absolute true value is never measured, data is used with the highest possible confidence of it being the most accurate description of meteorological conditions, with the least possible deviation from the truth. As described in 2.3.3, the data set best suited for this purpose is the re-analysis data set.

The specific data sets selected for this research were obtained from the Research Data Archive[2] made available by the National Center for Atmospheric Research in Boulder, Colorado. This archive holds data sets from a variety of international sources, such as National Centers for Environmental Prediction (NCEP), European Centre for Medium-Range Weather Forecasts (ECMWF), Canadian Meteorological Centre (CMC) and Japan Meteorological Agency (JMA). From among these, forecast data from data set *ds335.0* [103] was selected. Forecast data therein is gathered from a wealth of sources, from the likes of the itemized above.

For the re-analysis data, two data sets, namely *ds093.0* [132] and *ds094.0* [133], were ac-

[2]URL: `http://rda.ucar.edu/`

cessed to obtain required data. The data of the first data set stems from the Climate Forecast System Re-analysis (CFSR) program, provided by NCEP. However, the latest re-analysis was conducted only until December 2009 [100]. The CFSR has been conducted with the Climate Forecast System version 2 (CFSv2) model and has been extended from that date onwards by the CFSv2 [100, 134]. Therefore, for re-analysis data continuing from January 1st, 2011, the latter data set was utilized as a data source.

3.4. System overview

The proposed method closely resembles the schema suggested by Provost and Fawcett (see fig. 2.3), which foresees a data mining part to create a predictive model. In turn, this model (in this case the algorithm) is applied on a new data item. The result of this process herein is the forecast with a predicted deviation and an occurrence probability. Fig. 3.4 illustrates an overview of the proposed method, in which three main phases are pictured. The third phase represents the

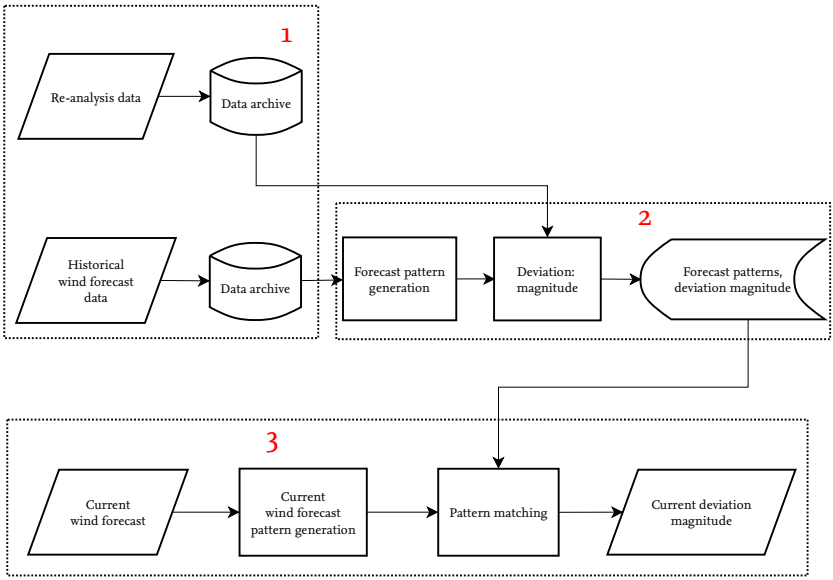


Figure 3.4.: System overview of the proposed method with three main steps illustrated.

usage of the algorithm (see dashed box no. 3 in fig. 3.4). A current arbitrary forecast

to be evaluated for predicted deviations serves as the input to the method. The arbitrary algorithm's knowledge is then applied onto this forecast, resulting in a revised forecast with the predicted deviations. To do this, the current wind forecast pattern is determined. A matching of this pattern to historical occurrences of this same pattern are retrieved, along with the deviations in wind speed that occurred at that point in time. Using this information, a likely deviation for the current wind forecast can be predicted, along with a value on the probability of the deviation coming into effect. This serves as the output.
Along with the application of the method's core algorithm, two other main phases have to be considered:

1. **Data storage, handling and tidying:** Weather data is typically delivered in GRIB2 files, which is a compact format containing raster data. While such a file is able to compress the final file size significantly, handling and processing of its data is difficult. A conversion to a more readable format is required for the data to be efficiently stored. (see dashed box no. 1 in fig. 3.4) Once this is done, the data will need to be coerced into a suited format to support the following data analysis. Chapter 4 elaborates in detail on the steps taken concerning data handling.

2. **Training of the machine learning algorithms:** After completion of the first step, the establishment and training of the algorithms can be performed. For this, historical forecast patterns are generated and the deviations between these and the *actual* reanalysis wind conditions determined. This information is then used for the training of all algorithms, yielding a model. Depending on the type of algorithm, this model may simply consist of coefficients of a Representing the algorithm's *learned knowledge*, it is then saved, as illustrated in box no. 2 in fig. 3.4. Chapter 5 describes the development of the algorithm in detail.

Hypothetical benefit of the proposed methodology to flight planning

The goal is to predict a magnitude of deviation to be expected in an arbitrary wind forecast as well as a probability of the respective deviation occurring. By identifying likely deviations, the uncertainty in wind forecasts is expected to be mitigated. As described in detail in 2.1.1, wind forecasts serve as an important input to any flight planning solution and can influence the planned path significantly. It is therefore hypothesized herein that a flight plan generated with a more accurate wind forecast may in turn result in it being generated with a lesser deviation from the actual flown route. Fig. 3.5 illustrates such a schematic view. A decline in discrepancy between the planned and the flown route underscores the increase in certainty that a

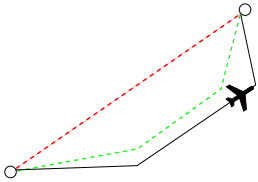


Figure 3.5.: Example for a deviation of the actual flown route to a flight plan with a greater deviation (dashed red) and one with a lesser deviation (dashed green).

flight will proceed as planned. Not only would this result in a higher predictability for airlines and potentially the airspace as a whole, but also in a reduction of the contingency fuel to be carried. This can be envisioned through a decrease of the forecast uncertainty. With the purpose of this fuel being to account for unforeseen meteorological conditions, the impact of which is assumed to be less with the utilization of the herein proposed method. The added certainty of a wind forecast may give pilots and dispatchers alike an augmented confidence, with which a smaller amount of contingency fuel may be justified.

3.5. Summary

This chapter first outlines a number of requirements for a system relying on Big Data machine learning to predict uncertainties in wind speed forecasts. According to these, the needed data is derived, including the types and geographical and temporal coverage. Lastly, the proposed system is presented as an assemblage of three main steps: data handling, algorithmic training/processing and application of the method.

4 Data processing and handling

This chapter describes the steps taken to ensure that the raw input data is converted and coerced into a suited format and structure for following data processing steps. In this chapter, the architecture of a Big Data cluster is first presented, in which data processing is performed. After describing this foundation, data processing to a suited working format are detailed. Specifically, details on the following necessary steps in preparing the needed data are outlined in this chapter:

- Data preparation process
    1. General Regularly-distributed Information in Binary form (GRIB)2 data conversion
    2. Data compression and upload to Big Data cluster
    3. Creation of external table
    4. Extract, transform, load (ETL) process: create *Parquet* table from external table
- Data tidying process
    1. Creation of temporary forecast and re-analysis data tables
    2. Joining of temporary tables into single, distributed *Parquet* table

4.1. Big Data cluster

The requirement for statistical relevance demands a large amount of data supporting the analysis. Such a large-scale data analysis prolongs the processing time. In order to still ensure that data processing is performed in a feasible time frame, a Big Data cluster is required. Such a system consists of a number of interconnected nodes, on which files can be stored in a distributed manner. On top of this, data analysis tools and query engines can be utilized to query the data, as described in 2.2.1. By relying on the open-source software *Hadoop* and the *MapReduce* paradigm [135], parallel processing and thus a greater level of processing efficiency can be realized.
The need for efficient storage and processing demanded the usage of a Big Data

cluster in this thesis. Usage of their cluster was granted by Boeing Research and Technology – Europe, with the physical system being located in Madrid, Spain. External access results in processing tasks being executed through remote access, with the entire processing running solely on the cluster itself. This yields the benefit that the user does not need to rely on a high-performance machine, as only the user's scripts are sent to the cluster and executed and the results of the data processing returned. Fig. 4.1 illustrates the entire user-cluster system utilized herein.

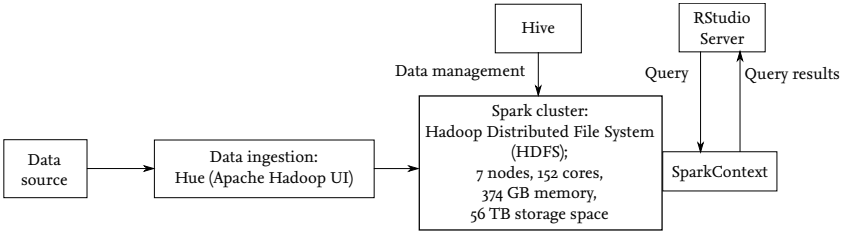


Figure 4.1.: Overview of the Big Data cluster with the key parts for data insertion, handling and analysis.

Data from any source is inserted into the Hadoop Distributed File System (HDFS) through *Hue*, the User Interface (UI) of Apache Hadoop. The HDFS is the core of the Big Data cluster and itself consists of 7 nodes, which all together yield a total storage space of 56 Terabyte (TB). For parallel processing purposes, the cluster provides 152 Core Processing Unit (CPU) and a maximum of 374 GB of memory. The number of cores can be selected depending on the amount of processing required. Memory usage depends on the framework used. While *MapReduce* relies more on disk, *Spark* aims to cache the dataset in memory, providing it fits. In all data analysis steps herein, the *Spark* framework is utilized.
The data warehouse software *Hive* is used to manage data that is stored in a distributed manner in the HDFS. The data preparation process relies heavily on this software. Data analysis is performed through a *RStudio Server* frontend, which translates the code written in the functional language *R* into Spark or MapReduce commands. In this way, the user can continue coding in a computing language that is more well-known and less complex than MapReduce commands. Additionally, *RStudio Server* provides the means to code and launch scripts remotely through browser access [136]. To launch and connect the RStudio environment with the cluster, the user needs to call a *SparkContext* and define the number of cores, executor instances and memory that is demanded for data processing.
The next sections will focus on the process of preparing the data for analysis, which

relies heavily on *Hive*.

4.2. Data preparation process

Data preparation is a necessary process to transform the raw data into a format with which it can then be tidied to eventually fit an optimal structure for data analysis. The raw weather data, as described in 3.3.5, is stored in the Research Data Archive, from which it was requested and downloaded. Using a software called *wgrib2*[1], the GRIB2 files were converted to Comma-separated Values (CSV) format. These first two steps are illustrated in fig. 4.2. CSV format requires significantly

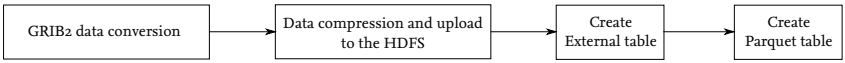


Figure 4.2.: Data preparation steps from acquisition and conversion of GRIB2 weather data to creation of *Parquet* tables.

more storage space. One month of forecasts amounts to around 450 Gigabytes (GB) of data. Even with a total storage space of 56 TB in the cluster, such storage requirements cannot be handled when considering that close to 10 years' worth of forecast and reanalysis data is to be stored. The CSV data was compressed using *Gzip* and uploaded via the Hue UI. Even with compression, storage requires approx. 21 TB of total space.
This storage requirement is another reason for restructuring the data. The goal is to eventually convert the data into *Parquet* format. This is a columnar format that provides high compression and performance with the Spark language, *SparkSQL* [137, 138]. To do this, in a first step, an external table needs to be created. This table does not store data, but merely references where the actual data rests on the system. Using this, the data can be read directly into a Parquet file. The ETL process for doing this in Hive is illustrated in algorithm 1.

Illustrated in alg. 1, two tables are first created. The first one is the external table, which has 7 variables defined and which carries the storage location of the compressed forecast data. In the next step, an empty Parquet table is created with the listed variables appearing as columns. Apart from the pressure level, validity time, longitude and latitude columns, all other variables are numbered as *0 through 4*. The reason for such a table layout is the temporal coverage of the forecast data. For a single validity time, up to 5 forecast are available, from 24 h prior to 0 h. As such, $UGRD_0$ is the column for UGRD values for 24 h forecasts, while $UGRD_4$ is

[1] `http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2/`

Data: Compressed forecast data stored in the HDFS
Result: Data coerced into Parquet file
Create *EXTERNAL TABLE*;
Variables: *creation time, validity time, variable, pressure level, longitude, latitude, value*;
Store as TEXTFILE;
Create *TABLE*;
Variables: *pressure level, validity time, creation times (0 through 4), longitude, latitude, UGRD (0 through 4), VGRD (0 through 4), HGT (0 through 4), TMP (0 through 4)*;
Store as PARQUET;
for *each data row entry* **do**
  if *Validity time* ***exists*** **then**
    Determine time step;
    Insert value into corresponding column;
  else
    Create new entry with validity and creation time;
    Insert value into corresponding column;
  end
end

Algorithm 1: Creation of an external table and loading forecast data into a Parquet file.

the column for 0 h forecasts/analyses.
After table creation, all data is parsed row by row. If a validity time step already exists, the valid creation time step is determined and the value inserted into the corresponding column. Should the validity time step not exist, a new entry will be inserted, with the process repeated until all rows are parsed.
Alg. 1 only illustrates the process for the forecast data. For the re-analysis data, the process is exactly the same, however with the difference that no indexes exist in the Parquet file. The reason for this is that the re-analysis is a report on the weather situation at one given time and location. Only one creation time can exist.

4.3. Data tidying process

Datasets with an explicit structure are called tidy datasets and can be easily manipulated and visualized [139]. In one, each variable is captured in a single column and each observation in a single row. GRIB2, as is common with meteorological data, is provided in a so-called *long-table* format. While such a format provides benefits in some applications, it cannot yield values of one variable being in one column. In order to ease further data analysis, the original GRIB2 structure needs to be transformed into a *wide-table* structure. The difference in structure is illustrated in fig. 4.3. The major benefit of a wide-table format is clearly evident: each column holds the values of one single variable. For the data in this research, this tidying process

Time stamp	Variable	Value
YYYY-MM-DD	UGRD	10
YYYY-MM-DD	VGRD	5
YYYY-MM-DD	HGT	100
YYYY-MM-DD	TMP	50

→

Time stamp	UGRD	VGRD	HGT	TMP
YYYY-MM-DD	10	5	100	50

Figure 4.3.: Structure of a long and wide-table format.

is performed in the steps outlined in alg. 2. First, the new single Parquet file is created with the appropriately named columns. In the following step, the forecast data is fully loaded to the table. The re-analysis data is then parsed row by row to find each row's appropriate validity time stamp in the single Parquet file. Once the correct row is found, the values are inserted into the corresponding columns.

Data: Forecast and re-analysis data in Parquet files
Result: Data joined into a single Parquet file
Create *TABLE*;
Variables: *pressure level, validity time, creation times (0 through 4), longitude, latitude, UGRD (0 through 4), VGRD (0 through 4), HGT (0 through 4), TMP (0 through 4), actual UGRD, actual VGRD, actual HGT, actual TMP*;
Store as PARQUET;
Insert forecast and re-analysis data;
for *each re-analysis data row entry* **do**
  if *Validity time* ***exists*** **then**
    Insert values into corresponding column;
  else
    Do nothing;
  end
end

Algorithm 2: Creation of a single Parquet file, which joins forecast and re-analysis data.

4.4. Summary

This chapter presents the process utilized in this thesis for data processing and handling. Outlined are steps that provide the means from converting the standard GRIB2 weather data format and coercing the data into a Parquet format structure on a Big Data cluster. This process is performed with the creation of an external table and an ETL process. In following steps, forecast and reanalysis data is further joined and tidied from two Parquet tables into a single structure for subsequent algorithm training.

Lat. [deg.]	**Lon.** [deg.]	**Pressure Level** [mbar]	**Date of validity**	**Date of creation (5 cols.)**	**Fcst: UGRD (5 cols.)** [m/s]	**Fcst: VGRD (5 cols.)** [m/s]	**Fcst: HGT (5 cols.)** [gpm]	**Fcst: TMP (5 cols.)** [K]	**RA: UGRD** [m/s]	**RA: VGRD** [m/s]	**RA: HGT** [gpm]	**RA: TMP** [K]

Table 4.1.: The joined final table structure.

5 Realization of a machine learning system for uncertainty prediction

This chapter focuses on the development of machine learning algorithms, which serve as the core of the system, illustrated as point 2 in fig. 3.4. Input to this part of the system represents the processed and cleaned data, as described in chapter 4. The goal of any machine learning algorithm is to recognize patterns from a set of training data and apply this to a new data instance. Besides the data volume involved, further levels of complexity are added by the possible temporal forecast steps and directions of wind concerned.
This chapter will first introduce a depiction of the major steps involved: training and testing of machine learning algorithms followed by a method of selecting a presumably optimal algorithm for generating a prediction, based on historical test data. Section 6 presents an evaluation and discussion of the algorithmic testing performed. Also being evaluated is the feasibility of the algorithm selection method.

5.1. Concept outline

The concept realized herein relies on three main components, as illustrated in fig. 5.1. The first part is detailed in the chapter on data handling, 4. That process' output is a data set in a format which can be utilized in the following two main components. These are first a stage for training and testing of machine learning

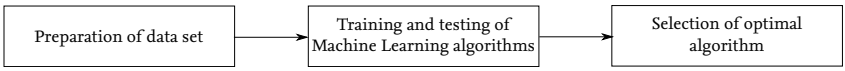


Figure 5.1.: Overview of the main components of the machine learning system.

algorithms, which is detailed in section 5.2. Included is a description of the logical structure and the functions and mechanisms utilized to ensure efficient data pro-

cessing. The second stage is the selection of the presumably optimal algorithm, given an input forecast. This is especially important, as no algorithm generates predictions which are always more accurate than the original forecasts. It is required to create a method that selects the appropriate algorithm, while relying on historical test data to support this selection. A detailed description of the underlying logic and realization is presented in 5.5.

5.2. Training and testing of machine learning algorithms

Training of machine learning algorithms is herein performed using the same data set for all algorithms. Specifically, a seed is set to ensure repeatability of the randomized selection process. Data valid at an arbitrary location is queried for and returned. Using the random seed, 90% of the data set is then selected and tagged as the training batch. The remaining 10% is allocated for testing purposes. This split is illustrated alongside the entire training process in fig. 5.2. The training

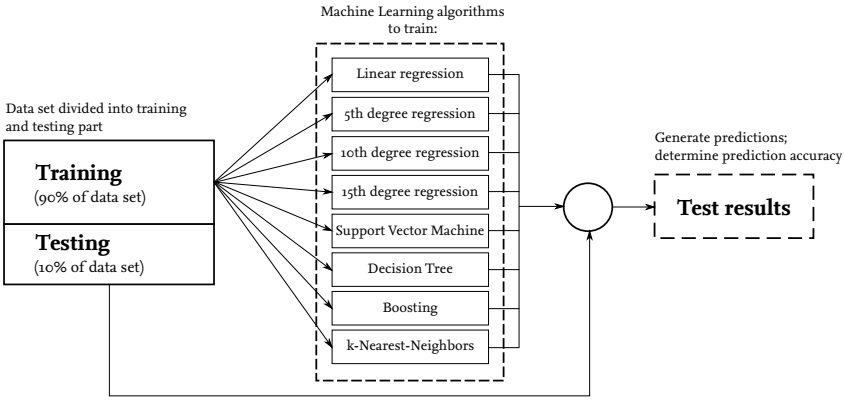


Figure 5.2.: Overview of the training and testing stage. Any data set is first split into a training and testing batch. The training batch is then utilized for training of machine learning algorithms, with the test batch eventually being applied to the algorithms. The resulting predictions are then compared to the actual values, with the accuracy being recorded. After completion of these two steps, the algorithms trained, as well as the test results (see dashed sections) are written to the HDFS for future usage.

data set is then applied to functions which train the respective machine learning

algorithms. A complete list of all algorithms trained, as well as the *R* functions and packages utilized is provided in table 5.1. A description of all algorithms is provided in chapter 2.2.3, which serves as the basis for derivation of algorithmic complexities in the following section 5.3. An explanation on the reason for the selection of these algorithms is provided in table 2.2.
After training, the test data set is then applied to every algorithm, in order to generate predictions. Present in the test set are both the original forecast, as well as the actual wind speed values. The discrepancies *original forecast – actual* and *algorithmic prediction – actual* can be determined in this process. The details of which are described in the chapter 6, prior to the evaluation of test results. The final step (illustrated in fig. 5.2 as dashed boxes) is the storage process to the HDFS of the trained algorithms on the one hand and the test results on the other. Trivially, the algorithms are stored to ensure future use. The test results are also stored, as these are needed in the selection method for the optimal algorithm, detailed in 5.5. The process illustrated in fig. 5.2 is repeated for both wind components, for all time steps and for all required coordinate locations. Each coordinate location is herein defined as a partition. Due to the necessary lateral and vertical coverage (in respect to a potential application in flight planning), the required number of partitions to process quickly exceeds the limits of what can be feasibly processed by a single computer. The *R* programming language, when used in standalone mode on a computer, needs to read into memory the data that is needed for algorithm training. Memory volume is limited by the computer's Random Access Memory (RAM). For each partition, two wind speed directions for five time steps need to be considered, resulting in 10 separate cases. For each of these cases, all algorithms are applied. Hence, a single partition necessitates the training of 80 algorithms. With a lateral resolution of $0.5°$ and 14 pressure levels, a total of 3,638,880 partitions around the globe need to be considered. Assuming that with a single computer, one partition can be processed at one time, with processing of one partition requiring 30 seconds, the total processing time required amounts to 1263.5 days. Yet this required processing is far from efficient. To ensure efficient processing, all processing is thereby performed in a Big Data cluster, the details of which are found in chapter 4.1. This allows distribution of the processing task onto up to 81 Core Processing Unit (CPU)s[1], as well as a means to parallelize it in part. This implementation is described in the following section.
Since algorithm training and testing is performed for a subset of partitions and not for all, the number of partitions to be processed in parallel is set at 284, dis-

[1] In theory, the number of CPUs can be further increased. In this setting however, the remaining cores (152 in total) are reserved for other processing tasks running in the cluster.

tributed across 49 CPUs. The time required to perform processing amounts to 12 minutes. This time includes all steps involved in the entire training and testing process, detailed further in alg. 3.

5.3. Computational complexities involved in training

The purpose of this section is to estimate the temporal effort needed in computing the algorithms herein. In light of ever-increasing computing performance, processing effort is commonly expressed as a function of various factors influencing the amount of needed processing, instead of the actual processing time. Hence, table 5.1 lists the algorithms concerned and their complexities in *big-O* notations. Also listed are the respective references of sources from which the complexities have been derived or cited from. The algorithms can be divided into four groups, based on their working similarities: regressions, Support Vector Machine (SVM), decision trees and forest and k-Nearest-Neighbors (kNN). Respective computational complexities are elaborated upon in the following.

5.3.1. Regressions

The method of Residual Sum of Squares (RSS) needs to be minimized in order to determine the β coefficients. To do this, this equation's first derivative is determined and set to zero, in order to obtain the unique solution [56]:

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y \tag{5.1}$$

$\mathbf{X}$ being the matrix of training data and y the predictant values. Eq. 5.1 can be divided into four operations:

1. $\mathbf{X}^T\mathbf{X}$: X is a matrix with n rows and d columns/dimensions. Hence, this matrix multiplication has a complexity of $\mathcal{O}(d^2 n)$.

2. Generating the matrix inversion of $\mathbf{X}^T\mathbf{X}$: For this operation, the number of data instances n do not influence computational complexity. Rather, it is only dependent on the number of dimensions in the data, $\mathcal{O}(d^3)$.

3. $\mathbf{X}^T y$: Since y is a vector with $d = 1$ dimensions, the complexity is reduced (in contrast to operation 1) to $\mathcal{O}(dn)$.

4. Computing the *LU/Cholesky* factorization to compute $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$: again, computational complexity is only dependent on the number of dimensions, $\mathcal{O}(d^3)$.

Algorithm	Function	R package	Arguments	Complexity	References
Linear regression	`lm`	`stats`	–	$\mathcal{O}(d^2n)$	Friedman et al. [56]
5th degree regression	`lm`	`stats`	–	$\mathcal{O}(d^2n)$	Friedman et al. [56]
10th degree regression	`lm`	`stats`	–	$\mathcal{O}(d^2n)$	Friedman et al. [56]
15th degree regression	`lm`	`stats`	–	$\mathcal{O}(d^2n)$	Friedman et al. [56]
Support Vector Machine	`svm`	`e1071` [140]	`eps-regression`, `radial kernel`	$\mathcal{O}(n^3)$	Tsang et al. [141] and Su and Zhang [61]
Decision Tree	`tree`	`tree` [142]	`recursive partition.method`, `deviance`	$\mathcal{O}(d^2n)$	Martin and Hirschberg [143]
Gradient Boosting	`gbm`	`gbm` [144]	`trees=100`, `Gaussian distrib.`	$\mathcal{O}(Td^2n)$	Martin and Hirschberg [143]
k-Nearest-Neighbors	–	–	`Euclidean distance`	$\mathcal{O}(1)$	Friedman et al. [56]

Table 5.1.: List of all algorithms trained, including the functions, packages and function arguments utilized. Each algorithm's training complexity is listed, as well as the sources from which the respective complexities have been derived. The logic for k-Nearest-Neighbors is self-programmed and is therefore not based on any package.

From these four operations, the first one dominates the other four. Hence, the linear regression's computational complexity is determined to be $\mathcal{O}(d^2n)$. These operations are also valid for any higher-degree polynomial fits, as the training data can be extended to $X_n = X_1^n$. Therefore, the only difference between n-degree fits to a linear one is that the training data needs to be multiplied with itself n times,

before these operations are performed. Hence, the computational complexity remains the same as with a linear regression, at $\mathcal{O}(d^2n)$.

5.3.2. Support Vector Machine

SVM training typically has a computational complexity of $\mathcal{O}(n^3)$. This is due to the underlying Quadratic Programming (QP) problem of the core SVM equation, 2.8, that needs to be solved. Essentially, the goal is to solve, for a given set of equations $\mathbf{A}x = b$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. [145] Underlying assumptions are that A is nonsingular, leading to the solution being unique for all b, as well as described by $x = \mathbf{A}^{-1}b$. Computation of the inverse has a complexity of $\mathcal{O}(n^3)$ [145], which also acts of the complexity term for a SVM.

5.3.3. Decision trees and forests

Decision tree building is described in 2.2.3. There, equations for both information gain and entropy are presented. While the information gain equation needs to be determined for every candidate dimension, the even more costly part is calculating $Entropy(\mathbf{S}_x)$, as an iteration through each candidate dimension is necessary. [61] This step yields a computational complexity of $\mathcal{O}(|\mathbf{S}|\, d)$, d being the number of dimensions. Additionally, unions of subsets at each stage of the decision tree require the total training data set $\mathcal{O}(|\mathbf{S}|\, d)$ with size n; the complexity for this being $\mathcal{O}(dn)$. The total computational complexity of a decision tree is $\mathcal{O}(d^2n)$.
Extending this knowledge onto a decision forest, in this case a Gradient Boosting algorithm, is trivially $\mathcal{O}(Tf)$. T represents the number of decision trees trained and f the effort needed to calculate a single decision tree. As it was shown that $f = \mathcal{O}(d^2n)$, the complexity of a forest is $\mathcal{O}(Td^2n)$. The number of trees is the only difference to the computational complexity of the single decision tree.

5.3.4. k-Nearest-Neighbors

kNN requires no training effort at all. The reason lies in the fact that the algorithm doesn't actually train a model that can be saved. Rather, the model is effectively the entire training data set and is only being used while in testing. The complexity for kNN training is $\mathcal{O}(1)$.

5.3.5. Dominating algorithm complexity

While a plurality of algorithms exhibit a complexity of $\mathcal{O}(d^2n)$, the Boosting algorithm is slightly more complex, as the number of decision trees built is an added complexity. The dominating algorithm to train is the SVM, as it features $\mathcal{O}(n^3)$.

5.4. Algorithmic implementation in the data cluster using SparkR

The logic behind the training and testing of all algorithms is detailed in algorithm 3. Central to this logic is the package SparkR v2.0. This version gives the user the ability to write user-defined functions and apply these in parallel on partitions of data. Fig. 5.3 illustrates the benefit of using parallelization in Spark. While a brute-force method would need to loop through all partitions one by one and needing the above-mentioned hypothetical processing time, the loop utilized in Spark ensures that a selected number of partitions are processed in parallel. The loop continues

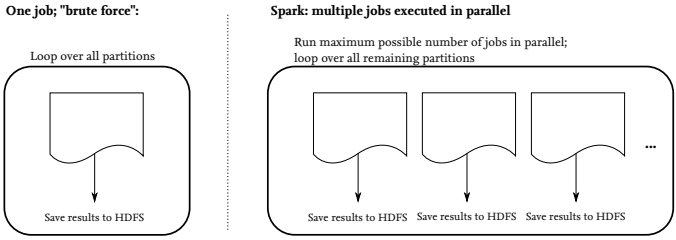


Figure 5.3.: Parallelization of tasks with Spark, as compared to a brute-force method, by Cabos et al. [44].

until all required partitions are processed. In theory, all partitions needing processing can be processed in parallel at once. Parallelization performance is limited by the number of available CPUs and the number of executors per CPU. With the maximum number of these set at 81 for the work herein, processing all partitions at once is not feasible, hence requiring a loop. Realizing this process with a loop also carries the benefit that in case of the job failing, all prior results are not lost. From the available SparkR functions that provide this parallelization, `gapply` as in "group apply", is chosen to be used herein. The reason for this is the possibility of the function to accept groups of data. Other functions provide the means to parallelize parts of lists. As the data herein is located in a single, large Parquet file, filtering for the required data is directly possible. Additionally, the data for a number of coordinates, and hence partitions, can be filtered for and then grouped by their latitude/longitude/pressure level value or key. In SparkR, such an object is called *GroupedData*, which can in turn serve as input to the `gapply` function. Algorithm 3 illustrates this logic applied for training and testing, starting with the

`Read.Parquet` function and the then following loop by longitudes. The reason for looping through longitudinal values lies in the fact that the Parquet file is itself partitioned by longitude. By calling a filtering function for a single longitude, processing time can be reduced, as only a single partition needs to be accessed. In the longitudinal loop, a filtering is then performed for data valid at latitudes for which algorithms are to be trained. `filter` creates a *SparkDataFrame*, which captures the data in Spark memory.
The filtering action is not directly performed due to Spark's lazy processing approach. Rather, all following required processing steps are first captured. In this case, this is the entire `gapply` function. After this function is recognized, will Spark actually perform the prior filtering function.
The same is valid for the `SparkR:::groupBy`, which groups the data in the SparkDataFrame by three variables, namely longitude, latitude and pressure level. This grouped data object then serves as input to `gapply`. As this function parallelizes all the partitions from the grouped data, it is important to note that all processing within this function is performed for every group. Two nested loops then perform the calculations for algorithm training throughout the data for all five time steps and two wind speed components. These are then in turn used together with the test data set to generate predictions. Calculations as to the accuracy of prediction and whether these are more accurate than the original forecast's are also performed, prior to eventual writing of all algorithms and test results to the HDFS.
In order to ensure that any stored data is identifiable, the `groupBy` key is carried along and attached to the two files per group. For this final process, the R package `rhdfs` is utilized, specifically its writing function, `hdfs.write`. This package, to the best of knowledge, serves as the only way to save data from a `SparkR` parallel loop. As this process works distributively, navigating to the user's local directory is not possible (at the time of writing). Instead, the data needs to be written to the HDFS directly, for which `rhdfs` provides the means.[2]

5.5. Selection of the optimal algorithm

This section presents the mechanism implemented herein to realize a selection of the presumably optimal algorithm, based on the test results generated in alg. 3. Commonly, the highest-scoring, or the algorithm with the greatest accuracy is chosen to be used. No algorithm is found to optimally generate predictions in all of the 4-dimensional (4D) space. Rather, algorithms are based on assumptions on how the data is structured. The prime example is linear regression, which assumes

[2] It is likely that in future versions of `SparkR`, its own native storage functions may be created.

```
Data: Weather data in Parquet format (see structure and variables in table 4.1)
Result: Machine learning algorithms trained, tested and saved to the HDFS
        together with the test results
Read.Parquet from "/HDFS/location";
Set seed(20);
for each longitude do
    SparkR:::filter for all data in required latitude range of single longitude;
    SparkR:::groupBy (key=) "pressure level", "longitude", "latitude";
    SparkR:::gapply on grouped data (per groupBy key) on
        Split into train set (90%), test set (10%);
        for all 5 time steps do
            for both wind speed components do
                Train Linear regression;
                Train 5th degree regression;
                Train 10th degree regression;
                Train 15th degree regression;
                Train Support Vector Machine;
                Train Decision Tree;
                Train Gradient Boosting;
                Train k-Nearest-Neighbors;
                Trim Algorithms of items not necessary for prediction;
            end
        end
        Generate test results/predictions with all algorithms using test set;
        hdfs.write all algorithms (with groupBy key) to storage in
         "/HDFS/storage/Algorithms";
        hdfs.write test results (with groupBy key) to storage in
         "/HDFS/storage/TestResults";
    end
end
```

Algorithm 3: The core logic involved in training and testing machine learning algorithms, including saving algorithms and test results to the HDFS. The R package SparkR ensures parallelization.

that linear relationships exist in data. Due to these assumptions, each algorithm will generate good predictions in some areas, while in others, the underlying assumption does not at all model the data well. An exemplary figure illustrating different parts of a distribution inhibiting different best-performing algorithms is provided in 5.4. Because of this phenomenon, further amplified by the large

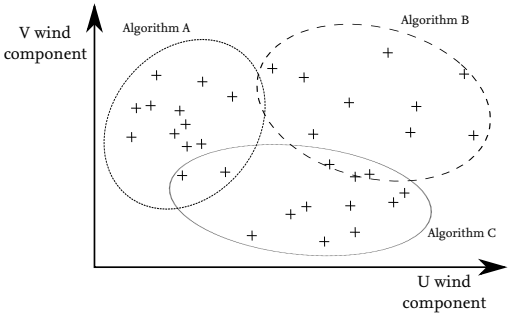


Figure 5.4.: Exemplary possible spread of test data with the respective best-performing algorithms indicated.

backlog of historical data herein and the dynamic properties of weather in general, simply selecting the overall best-performing algorithm will not be a fitting solution. Instead, by determining first which data points in the backlog are most similar to the current new forecast item (on which a prediction is to be generated), the *vicinity* in terms of algorithm performance can be first gauged. Hereby the hypothesis is that similar forecast items in history were predicted well with a given algorithm, thus this new forecast item will also be predicted well using the same algorithm, due to the forecast's similarity to those historical points. In order to realize this idea in this context, an algorithm selection process will have to determine one point's similarity to all historical points, choose a given k nearest/most similar point(s) and then determine what algorithm was the best performing in this k-th vicinity.

The selection process devised is illustrated as a flow diagram in 5.5. At this stage, the test results have been generated in the previous process and have been saved to the HDFS. Also, a new forecast item is to be applied to an algorithm to generate a prediction on the uncertainty of this new forecast. The process begins by loading both the new forecast item and the test result matrix. As already described in the prior paragraph, a metric to determine similarity is needed. For this, Euclidean distance is calculated (the definition can be found in chapter 2.2.3 for k-Nearest-

Neighbors) including a normalization step for all 4 dimensions. Following this step, a distance matrix is generated in the structure of a single row (assuming a single new forecast) and m number of columns for the number of data points in the test result matrix. Using this distance matrix, the $k = 6$ nearest points can be determined. These are then used in a voting scheme which then outputs the presumably most optimal algorithm, based on similarity. Finally, this algorithm is used for prediction on the new forecast item.

The main mechanisms of this process, namely selection of the nearest data points

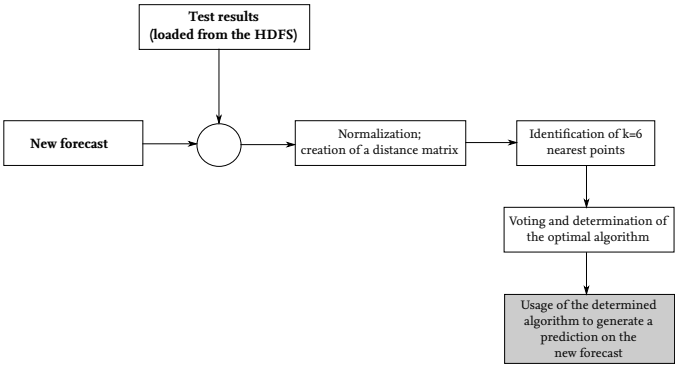


Figure 5.5.: Flow diagram of the algorithm selection method.

and the subsequent voting for the optimal algorithm, is further detailed in fig. 5.6. A number of $k = 6$ neighbors are searched for, with this number set arbitrarily. If set too high, the points selected may not be similar enough and set too low, the method may be prone to bias.

In fig. 5.6 in the top left, the new forecast item is illustrated as a the center black dot. The surrounding six points indicate the 6 closest data points in terms of Euclidean distance. In a next step, the respective best-performing algorithm and the distance to the new forecast item of each of these six are gathered in a matrix. Using this information, voting weights can be allocated. Each neighboring data points receives a numeric weight per the distance it is located away from the new forecast. If it features no distance at all, it receives the strongest weight of 1, whereas it is located at a distance of 1, the data point receives zero weight. These values per the respective algorithm are summed up to generate the bottom left graph in fig. 5.6. The algorithm with the highest score is then selected to be used for prediction on the current new forecast item.

Additionally, some rules apply in this logic in order to handle special situations

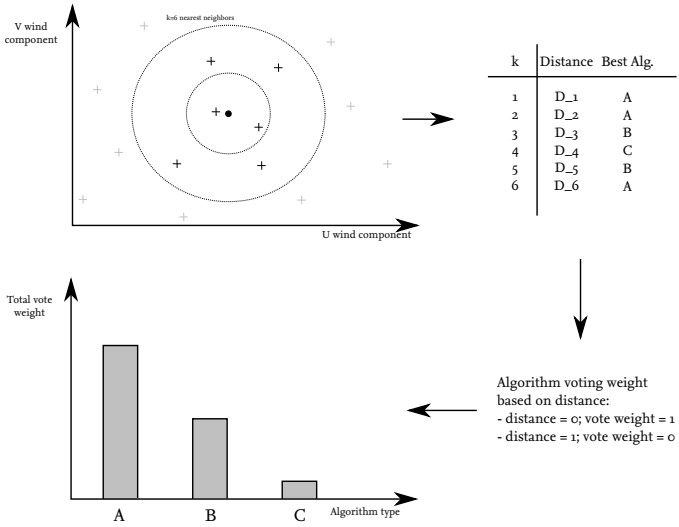


Figure 5.6.: The underlying mechanisms for voting and determining the best-performing algorithm, for 6 exemplary data points.

in cases in which neighbors do not feature a single algorithm that generated a prediction with lesser error than of the original forecast's. More technical and algorithmic details of the method are presented in algorithm 4, while the rules are discussed afterward in more detail .

The core algorithm selection logic described in 4 uses the same mechanisms as the logic for training machine learning algorithms, as shown in 3. Processing relies on the `SparkR:::gapply` function receiving grouped data. This data needs to be from a separate set and must not feature in both training and test data sets. In the `gapply` function, each group's location key is first determined, which is then used to retrieve the output files from algorithm 3; the trained algorithms and test results. As these need to be imported from the HDFS and not from local storage, the package `rhdfs` with its `hdfs.read` function is employed. After this step is complete, all machine learning algorithms are effectively present in the loop's workspace as common R objects. The then following steps are all repeated for all five time steps and both wind speed components, effectively yielding 10 different combinations, e.g. *"U wind speed component, 24 hour forecast steps"*. The input validation data is filtered for each respective combination and normalized together with the values in the test results matrix. Normalization yields numbers between 0 and

Data: Weather data in Parquet format from the validation data set
Result: Nearest historical points found, voting for optimal algorithm performed and selected algorithm used to generate prediction on current forecast item
Read.Parquet *from* "/HDFS/location/of/validation/dataset";
for *each longitude* **do**
  SparkR:::filter *for* all data in required latitude range of single longitude;
  SparkR:::groupBy (key=) "pressure level", "longitude", "latitude";
  SparkR:::gapply *on grouped data (per groupBy key)* **on**
    Identify current location key: *latitude, longitude, pressure level*;
    hdfs.read all algorithms & test results with current location key from HDFS;
    for *all 5 time steps* **do**
      for *both wind speed components* **do**
        Normalize forecast items and data in test results;
        Generate distance matrix of n forecast items and m test result instances;
        for *every forecast item* **do**
          Find $k = 6$ nearest points in distance matrix, distribute voting weights per distance;
          Aggregate voting weights, determine spread of vote weights among algorithms;
          Apply rules;
          Generate prediction using selected algorithm;
        end
      end
    end
  end
  Write predictions to Parquet format file on the HDFS;
end

Algorithm 4: The core logic involved for selection of the optimal algorithm per forecast item. A detailed description of rules is provided in algorithm 5.

```
if at least 1 nearest point has no prediction better than its original forecast then
    if nearest neighbor AND ≥ 1 neighbors feature no algorithm then
        Do not generate prediction AND keep current forecast;
    else if second nearest neighbor AND ≥ 2 neighbors feature no algorithm then
        Do not generate prediction AND keep current forecast;
    else
        if two or more algorithms have the same sum of votes then
            Select the algorithm for which the nearest point voted for;
        end
    end
else
    if two or more algorithms have the same sum of votes then
        Select the algorithm for which the nearest point voted for;
    end
end
```

Algorithm 5: Rules by which algorithms are selected, for the process detailed in algorithm 4.

1. A distance matrix based on Euclidean distance is then generated, in which the number of validation data set instances are ordered in the rows and the test result instances in the columns, as illustrated in fig. 5.6. By doing so, each row holds the distance values for that respective validation data instance to all other test results instances.

A loop then performs the algorithmic selection separately for each point (or row in the distance matrix), by first determining the 6 closest points (or columns in the distance matrix) of each validation data point. Each one of these six points' best-performing algorithm is then determined and voting weights allocated, based on their respective distances. As indicated in fig. 5.6, an identical point (i.e. one with a distance of 0) would receive the maximum vote weight of 1, whereas a very distant point with a distance of 1 would receive a vote weight of 0. It is possible that distances of >1 are calculated. In that case, the voting weight received would also be 0. After voting, the weights are summed up for all algorithms that were voted for and the spread of voting weights determined.

At this stage, a number of checks have to be applied to identify three scenarios that require pre-defined actions. The first handles cases in which algorithms receive the same numerical vote values, while the two others apply for cases in which points are identified, for which no algorithms were able to generate predictions

with a higher accuracy than of that point's original forecast.

1. **Two or more algorithms have the same total numeric vote.** While rather unlikely, it is nevertheless theoretically possible. Should this scenario occur, the algorithm of the nearest point is selected, provided this point's accuracy was improved with this algorithm. If this is not the case, the next-nearest point is then chosen.

2. **The nearest neighbor and ≥ 1 other neighbor(s) feature no algorithmic prediction improvement.** In this situation it is assumed that as the nearest and at least one other point does not have at least one algorithm that generated a more accurate prediction than of that point's original forecast, it cannot be ensured that any of the algorithms will generate a favorable prediction for the current validation forecast item. In essence, in such a situation the vicinity of the current validation point would be effectively deemed to be unsuited for accurate algorithmic predictions. This logic would demand that instead of generating an unfavorable prediction, it is safer to retain the original forecast value instead.

3. **The second nearest neighbor and ≥ 2 other neighbors feature no algorithmic prediction improvement.** This situation resembles the second scenario, however with a greater number of neighbors. The underlying assumption is that as the nearest neighbor still produces a vote for an algorithm, a second-nearest neighbor featuring no algorithmic prediction improvement can be tolerated. If too many other points also do not feature an algorithmic improvement, the reasoning is to rather rely on retaining the original forecast value instead.

Rules for these three scenarios are realized in the core logic (see alg. 4) as *if/else* clauses. Once these have been applied, in the case of an algorithm being chosen, this algorithm will then be used to generate a prediction on this current forecast item. In case of an identification of scenario 2 or 3, the original forecast value is retained and no algorithm is used for prediction. This process is repeated, as stated above, for all time steps and both wind speed components. Finally, the results are saved to a Parquet file on the HDFS.

5.5.1. Computational complexity of the algorithm selection method

This section elaborates on the computational complexities involved in the algorithm selection method. Specifically, the complexity is described for the validation

process, described in detail in section 6.2. The complexity involved in utilizing this process on a continuous basis in regular intervals (e.g. every 6 hours) is also determined.
To estimate computational complexity, the following variables are first defined: n describes the number of validation data set instances, d the number of dimensions, a the types of algorithms, p the number of data partitions, t the number of time steps and m the number of test result instances. As illustrated in fig. 5.5, the algorithm selection process consists primarily of four steps after both validation data and test results have been loaded. The computational complexities of each step are determined using these defined variables in the following:

1. **Normalization of test results and validation data set instances**: The test results matrix holds m instances, of which each exhibits a constant $d = 4$ dimensions. The same number of dimensions trivially apply to all n validation data instances. Normalization requires that a maximum and minimum value be found in one data set, with which the normalization process is performed. Hence, the searching of these values in the test results set requires passing by $m \times d$ times. After determination of these two values, the normalization itself needs to be performed across all $d = 4$ dimensions of all $m + n$ instances. The dominating factor concerning complexity is $\mathcal{O}(d(m+n))$.

2. **Creation of the distance matrix and identification of $k = 6$ nearest points**: The distance matrix is of size $n \times m$, as the distance between every nth validation to every mth test result instance needs to be calculated. Since the distance is a function in d dimensions, the effort needed equals $\mathcal{O}(dmn)$. For all n validation data instances, all m columns of the distance matrix have to be processed once more, in order to determine the $k = 6$ least distances (for every nth instance). These two processing tasks together yield a complexity of $\mathcal{O}(dmn + mn) = \mathcal{O}((d+1)mn)$. Due to the fact that d is constant and of a small number ($d = 4 \ll n, m$), the term $(d + 1)$ can be ignored, yielding a computational complexity for this step of $\mathcal{O}(mn)$.

3. **Voting and determination of the optimal algorithm**: In this step, the $k = 6$ nearest neighbors are assumed available from the previous step. For every instance, the prediction performance of the $a = 8$ different algorithms needs to be retrieved, yielding a constant complexity of $\mathcal{O}(ka) \rightarrow \mathcal{O}(1)$. This means that this process takes a constant time.

4. **Usage of the determined algorithm**: This last step only involves one algorithm at one time and repeated n times for different algorithms. Of the eight algorithms trained herein, the following respective test complexities exist:

- **Linear and higher degree polynomial regression**: Since the trained algorithm is essentially a number of coefficients, regression algorithms are used by inserting the data instance's values. This is not dependent on the degree of the algorithm and therefore yields a constant complexity of $\mathcal{O}(1)$.
- **Support Vector Machine**: A trained SVM has n_{SV} number of trained support vectors, which feature in d dimensions. Applying a data instance means iterating through all support vectors, in all dimensions; the aim being to find the closest vector. Therefore, the complexity of using an SVM is $\mathcal{O}(n_{SV}d)$.
- **Decision tree**: When using a decision tree, the main task is to find the leaf the data instance falls into by performing comparisons. The number of constraints are a function of the number of levels in a decision tree. Hence if h is the height of a tree, the complexity to find the fitting leaf is of the order $\mathcal{O}(h)$.
- **Gradient Boosting**: As mentioned in section 5.3, Gradient Boosting algorithms are linked to decision trees, as they create a multitude of trees to function. It uses these trees to iteratively improve on the last tree and yielding a single tree structure. Therefore, as with a decision tree, the complexity is $\mathcal{O}(h)$.
- **k-Nearest-Neighbors**: When using kNN, this algorithm only actually performs calculations. kNN primarily determines the k nearest, or most similar neighbors. To do this, a matrix describing distances between n validation and l test results instances needs to be determined. Prior to generating the distance matrix, the data needs to be normalized. The computational complexities for these two operations are the following:

  a) Normalization: Since both data sets need to be normalized, $n + l$ number of data instances need to processed. Assuming that the data inhibits $d > 1$ dimensions, the resulting complexity hence equals $\mathcal{O}(d(n + l))$.

  b) Distance matrix generation: As the distance between l test results and n validation instances are to be calculated, a matrix of size $l \times n$ is hence created, yielding a complexity of $\mathcal{O}(nl)$.

  Of these two operations, the second one clearly dominates, under the condition that $d \ll n, l$. Hence, the computational complexity for kNN is determined to be $\mathcal{O}(nl)$.

By comparing the various complexities it can be observed that the second step, i.e. the creation of the distance matrix and the searching of 6 nearest points, is the dominant computational complexity when performing the validation, at $\mathcal{O}(mn)$.

5.5.2. Complexity considerations when employed in a real-world setting

When considering this algorithm selection process as a real product, regular runtime intervals have to be considered. This is due to the fact that weather forecasts applicable to the method herein are generated every six hours. At every six hours, this process is needed to be initiated. For this step, the actual upper bound complexity is to be evaluated.
In light of the maximum complexity for each initiation step, it is assumed that the complete number of partitions is $p_{max} = 3,638,880$[3]. Unchanged in regards to section 5.5.1 are the types $a = 8$ of algorithms, n validation data instances, $t = 5$ time steps and m test result instances. With each initiation, the process illustrated in fig. 5.5 is hence repeated $p_{max} \cdot t \cdot n = 18,194,400 \cdot n$ times. At this point, a major difference exists between actual real-world application and bulk validation. With every 6-hourly step, only a single forecast/instance is evaluated, contrasting the single bulk validation run. Thus, $n = 1$, reducing the necessary repetitions to $p_{max} \cdot t = 18,194,400$ times per run.
This result indicates that when real-world application of this process is considered, the computational complexity is constant, i.e. of order $\mathcal{O}(1)$. One single exception to this is the case in which the test results data set continuously grows. It can do so every time a forecast is evaluated, its growth thus being linear: $\mathcal{O}(m)$. This would affect the second step in 5.5.1, as its complexity is dependent on m: $\mathcal{O}(mn)$. Therefore, even if the test results data set grows over time, the algorithm can nevertheless still be solved in linear time.

5.6. Summary

This chapter elaborates on the realization of the proposed concept. Firstly, the three main logical steps needed for this are outlined, before focusing in detail on the training/testing of algorithms and a proposed method to select the most optimal algorithm, based on historical data. For the prior, a number of machine learning algorithms and the architecture embedding these are presented. Following this, the focus is laid on the computational complexities involved for this step. These are separately discussed for training and testing, per algorithm. Af-

[3]Considering a lateral resolution of 0.5° and 14 pressure levels. See section 3.3.2 for more details.

ter this, the actual implementation and execution in the data cluster is described. The chapter then finishes with a detailed outline on the algorithm selection process, the underlying logic, implementation and the computational complexities involved.

6 Evaluation of concept

This chapter presents and discusses the results of two evaluations performed, to validate algorithmic prediction performance. Fig. 6.1 illustrates on a high level the processes detailed in this chapter, in order to explain the reasoning behind dividing this evaluation into two tiers. The concept realized in 5.1 is depicted at the top left, with the training and test databases, as well as the training and test process. Further, the process for selecting the optimal algorithm based on input data, specifically explained in 5.5, is shown on the right of fig. 6.1. As with any data

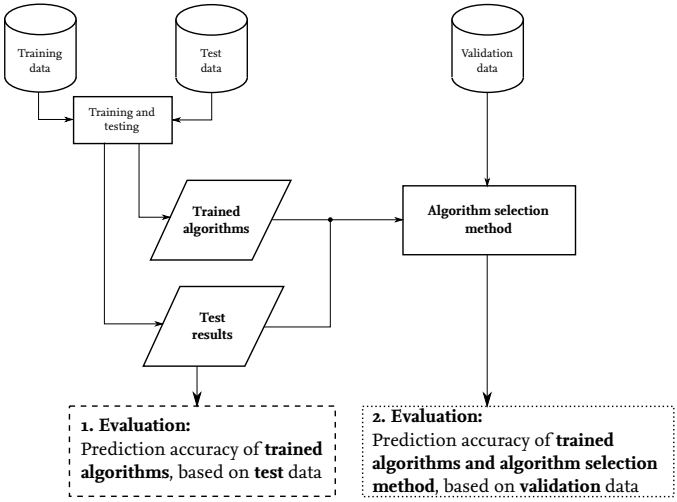


Figure 6.1.: Overview of processes, generated data and two evaluation steps: first, an evaluation of algorithmic accuracy with a test data set; followed by an evaluation using a validation set, to validate the prediction performance added by the algorithm selection method.

analysis and algorithm training process, a post-analysis has to be performed to test the trained algorithms' prediction accuracy or performance. This is accounted for in the first evaluation, for which the test data is utilized (see section 5.2 for details on creation of the training and test data sets). All trained algorithms are used to

generate predictions on this test set, the results of which are then presented and discussed in section 6.1.
This evaluation step is not sufficient to validate the entire concept. Due to the second process, in which the optimal algorithm is to be identified with usage of prior test results, another evaluation step is needed to validate the performance of this process. For this evaluation, a validation data set with entirely new data, is employed. The goal of this second evaluation (see section 6.2) is to evaluate the prediction performance of trained algorithm, with utilization of the selection process, on new validation data.
This chapter concludes with a sensitivity analysis for the algorithm selection method. Due to computational complexity, it is limited to two locations that feature major meteorological differences.

6.1. Test set evaluation of trained algorithms

This section presents and discusses the results from the evaluation of the trained algorithms using the test data set. First, a number of hypotheses are proposed, after which the results are discussed, leading to a falsification or no falsification of these. Finally, run times for the algorithmic training is provided.

6.1.1. Hypotheses

For the test set evaluation of trained algorithms, the following hypotheses have been proposed:

1. **Hypothesis H1:** If trained machine learning algorithms are used to generate wind speed predictions, then coherent geographic patterns of best- and worst-performing algorithms are retrieved.

2. **Hypothesis H2:** If trained machine learning algorithms are used to generate wind speed predictions, then at least one algorithm's Mean squared error (MSE) will be lower than the original MSE exhibited by the original forecast at every location.

3. **Hypothesis H3:** Each algorithm's MSE will increase with decreasing forecast lead time, irrelevant of the location.

6.1.2. Outline of test evaluation

Due to the complex nature of the analysis, this evaluation chapter is divided into a number of sections. The primary interest of which is to analyze the **prediction performance** and **robustness** of all algorithms. Making matters complex is the need to

perform this evaluation for each algorithm on both wind speed components (U and V), all processed lateral coordinates, all five time steps (from 24h to 00h forecasts) and lastly, all vertical/pressure levels. Therefore, a first section (6.1.3) is dedicated entirely on prediction performance of algorithms throughout all named dimensions, except pressure levels. Instead, a level at which flights commonly cruise is chosen to be analyzed. In a second section (6.1.5), algorithmic robustness is the focus. As in the prior section, results are discussed for all dimensions for a single pressure level. This last remaining complexity will then feature in the final section 6.1.5, in which a summary of prediction performance and robustness throughout the processed levels is provided.

6.1.3. Algorithmic prediction performance

When set into the context of machine learning, desired algorithmic performance can be expressed as a lesser discrepancy between a true value and the prediction than of true value and the original forecast. Mathematically, this can be expressed as:

$$\Delta_{AlgorithmicPrediction} < \Delta_{OriginalForecast} \tag{6.1}$$

This metric, effectively describing the prediction error, will feature throughout the entire evaluation of test results. However, the simple mean of prediction errors will be avoided. The reason for doing so is the fact that prediction values can be both greater and lesser than the true value, yielding positive and negative values. These two will neutralize each other in part and thereby leading to falsification of the result. The more reliable MSE is relied on:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y}_i - Y_i\right)^2 \tag{6.2}$$

With $\hat{Y}_i$ expressing the prediction of an arbitrary algorithm and Y_i being the true value. Calculating the mean of the squared instead of the normal error provides the benefit that all values are positive. One other notable effect to take into consideration is the MSE's greater punishment of larger errors due to the squaring of the difference. Nevertheless and because of the prior reason, the MSE is heavily relied upon in the course of this work.

As indicated in 6.1.2, the evaluation of test results in this section will be performed on a single pressure level. In a prior paper [44], the pressure level equalling 200 Millibars (mbar) is used as its altitude of 36,000 ft in the International Standard Atmosphere (ISA) is a common range for cruising flights. For this reason, this pressure level is also retained for the evaluation in this section. Moreover, the processing of locations is limited in general to four pressure levels from 150 through

300 mbar and the area pictured in fig. 6.2. This area consists of a rectangular part

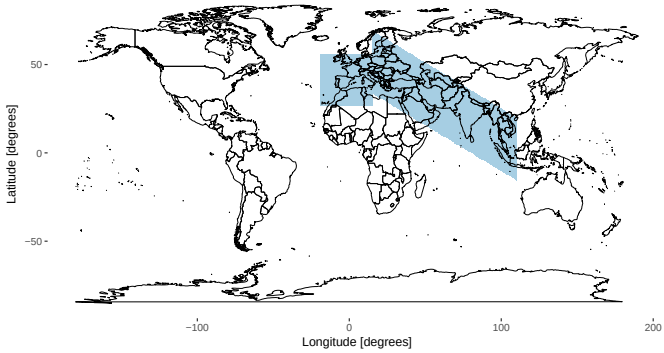


Figure 6.2.: Lateral extension of coordinate locations processed colored blue.

in between 19° West to 15° East and 27° to 55.5° North. A second area stretches from 15.5° to 110° East. This area can best be described by an interpolation between two coordinates (N44° E15.5° and S5° E110°), yielding all coordinates in between. By adding 10 lateral coordinates south and 25 north of this line, an area resembling a parallelogram is established. The decision to process these areas is driven by the following usage of forecasts in these regions to generate flight plans. Selected flights between the Canary Islands, Germany and South East Asia thereby demand the processing of the selected area in fig. 6.2.

6.1.4. Processing times for training and testing

Training and testing over the data set's coverage outlined in section 6.2 is performed using 88 Core Processing Unit (CPU)s in the data cluster. Instead of processing all partitions in parallel at one time, the code is applied in iterations. One iteration performs processing on all partitions of a single longitude. Thereby, starting at 341° and ending at 110°, in total 259 iterations are required. This entire process requires a processing time of 5 days and 7 hours. Assuming a hypothetical processing time of 30 seconds per partition, in comparison, such a brute-force looping would require 24 days and 7 hours.

6.1.5. Evaluation of algorithmic MSE

This evaluation will first focus on the linear regression algorithm applied on both wind speed components for the 24 h time step on the 200 mbar pressure level. Re-

sults from the test run are presented and discussed thereafter. This process will then be extended across all time steps. Results from the other seven algorithms are presented and discussed afterward. A summary is then provided for an aggregated discussion across all algorithms.

Results for a linear regression on 24 h forecast data In the case of testing a linear regression algorithm, fig. 6.3 illustrates the results found. These results are

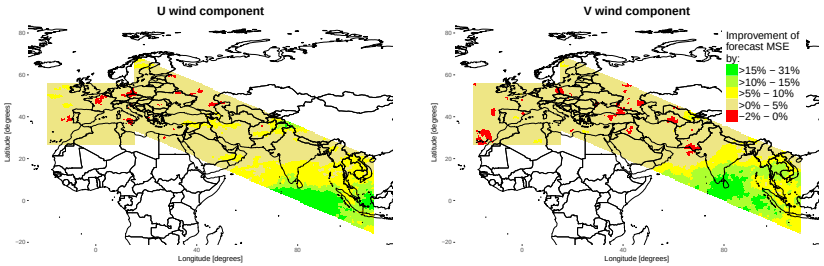


Figure 6.3.: MSE test results for the linear regression, applied on data at 200 mbar altitude, for U and V wind components.

expressed in improvement over the original MSE value of the test data and normalized in percentage values. Notably dominant for both wind components is the prevalence of the largest region of $> 0 - 5\%$ improvement. Throughout the region covering western Europe to the Caspian Sea, scattered patches of a decrease of U forecast MSE can be observed. For the V component, these feature across a greater area, with the easternmost situated on the Pakistani/Indian frontier. Notably absent from areas west of the Indian subcontinent are areas featuring an improvement of greater than 10% for both wind components. MSE results greatly increase across India and South East Asia, harboring the greatest improvement numbers of up to 31%. For the U wind component, these are found over the northern Indian Ocean and parts of the South China Sea. While prediction performance is generally high over the Indonesian islands, this is not the case for the greater part of the Malaysian peninsula. These patterns are different when the V wind component is concerned. While the northern Indian Ocean features the greatest improvements as the U wind component, percentages in South East Asia generally fall into the mid-range of $> 0 - 5\%$ improvement.
Results from both wind components do not show boundaries and/or patterns corresponding to any geographic contours or aspects, such as coastlines or mountain

ranges. Instead, the results form clear patterns of coherent prediction improvements.

Discussion of 24 h linear regression results Three observations in general are of particular interest: clearly, prediction performance of a linear regression is bound by the region the data is valid at, geographic patterns do not influence algorithmic prediction performance through skewed data and lastly, lateral patterns of differing performance can be observed.
The reason for the first observation may lie in the fact that weather forecasting is generally more difficult in the equator region. There, more volatile weather oftentimes leads to an increased difficulty when predicting weather phenomena. The result is a lesser forecast accuracy than for ones in central Europe. Lesser forecast accuracy in turn allows more potential for improvement. The linear regression's predictions may draw a benefit from this situation, which may result in the high percentage improvements over the original forecasts. Observable patterns of areas with coherent improvement percentages, without the showing of geographic contours might be explained by the altitude at which the data is valid at. A pressure level of 200 mbar equals an altitude of approx. 36,000 ft in the ISA, which is significantly higher than all mountain ranges, even the Himalayas. An influence of even these mountains onto the raw data cannot be observed, as is evident in fig. 6.3. The reason may lie in the fact that the distance between the mountain range and 200 mbar is too great for any weather effects to be traced in the data.
The observed patterns of coherent improvement percentages can be interpreted as a sign of algorithmic stability. As the patterns exhibit clear contours, a potential null hypothesis that lateral improvement performance is random, can be falsified for this case. Also speaking for coherency is the fact that areas of different improvement consistently border either the next higher or lower improvement category. This phenomenon is well-observable in fig. 6.3 for the V wind component in the areal vicinity of southern India/Sri Lanka. The area of greatest improvement ($> 15\% - 31\%$) is entirely surrounded by the next best ($> 10\% - 15\%$) while that in turn only borders points featuring an improvement of $> 5\% - 10\%$.

Results of linear regression throughout all time steps The results of applying a linear regression on the same data throughout the remaining four time steps are illustrated in fig. A.1; whereby the 18 h time step is the top and the 00 h/analysis step is the bottom row. The macroscopic patterns found in fig. 6.3 for the 24 h step are also found throughout the remaining four. This large area south of India and covering parts of South East Asia is also reflected for both wind components. Locations west of the Central Asian republics also feature similar results as in the 24 h time step: large areas of $> 0 - 5\%$ improvement over the original forecasts, with

limited patches of negative improvements scattered throughout. These patterns are prevalent throughout all time steps, albeit with a greater occurrence frequency when comparing the 24 h and 00 h time steps for the V wind component.
One notable consistency throughout 24 h to 06 h forecasts for the U wind component is an area in north Pakistan/India of with high local improvement numbers. These, with up to 31% improvement, are in contrast to the regional results which mostly lie in the $> 0 - 5\%$ improvement range. Throughout the time steps, these results remain consistent both in the percentage of improvement and its lateral coverage. Only in the 00 h time step does this pattern diminish. A trend can be identified regarding the maximum improvement. For the 24 h time step, the maximum is set at 31%, which decreases consistently to 25% for the 00 h time step. On the other hand, not only the greatest prediction MSE improvement is of interest, but the number of coordinates that feature an improvement over the original forecast MSE. These numbers have been illustrated in fig. 6.4. A number of trends can

Figure 6.4.: Number of coordinates in each of the five improvement categories for a testing of a linear regression for both wind components throughout all time steps, at 200 mbar altitude. Forecast lead times range from 24 h (leftmost column) to 00 h (rightmost column).

be retrieved from fig. 6.4. For both wind components, the number of coordinates in the two greatest improvement categories decreases with a decreased forecast lead time. The inverse trend is valid for the group of $> 0 - 5\%$ improvement, which steadily grows with a declining lead time. The remaining two categories behave differently. While steadily increasing in number with the V wind component, the number of worst-performing coordinates actually declines between the 06 h and 00 h forecast step of the U wind component. This same trend is observable for the third-best performing category. For both wind components, the number of $> 5 - 10\%$ coordinates first increases, before decreasing again between the 06 and 00 h steps.

Discussion of linear regression test results throughout all time steps In general, the improvement results retrieved for the application of the linear regression on all time steps mimic the findings already drawn for the above 24 h results. Throughout all time steps and both wind components, the areas with best forecast improvement are found in the southern India/South East Asia region. As with the 24 h time step, the reason may be once more that the original forecast carries a lesser accuracy than with forecasts in e.g. western Europe. The second noticeable trend throughout all time steps is a decrease in the maximum improvement percentage, a decrease in the total number of most-improved coordinates and an increase in the lesser to not improved coordinates. An explanation to this phenomenon may simply be that the lesser the forecast lead time, the greater the inherent forecast accuracy. With a decreasing forecast error, the potential also decreases for an algorithm to generate a prediction that actually is closer to the true value.

Results of higher-order polynomial regressions Results from higher-order polynomial regressions (see fig. 6.5) resemble in some aspects the findings recorded in fig. 6.3 with the linear regression. Noteworthy are the vast areas in southern and South East Asia, which generally score high improvement levels of up to 32%. Throughout all results of 24 h U wind regressions (left column in fig. 6.5) the high-performing area identified over northern Pakistan is consistent with prior results from the linear regression. Also, improvement number patterns of each wind component stay consistent in size, coverage and location throughout all regressions. While the maximum improvement of the original forecast MSE is unchanged when compared to that of the linear regression, negative improvements escalate for the 10th and 15th order polynomial regressions to a maximum of over 43 million percent. Also appearing for the two highest-order regressions are scattered points with very great negative improvements throughout the entire area processed and for both wind components.

Discussion of higher-order regression results While the results mimic those from the linear regression quantitatively and qualitatively, two major differences exist: very high decreases in forecast accuracy, i.e. deteriorations of forecast accuracy and scattered points where the linear regression performed well. When increasing the order of polynomial regressions, the swings in the fit too increase. The regression fit will also grow towards either positive or negative infinity on either end of the d-dimensional space the data is occupying. Any test data point in these regions will incur significant prediction error, which is only amplified with the growing order of regression. This is the explanation to the high negative improvement values in the 10th and 15th order regressions.

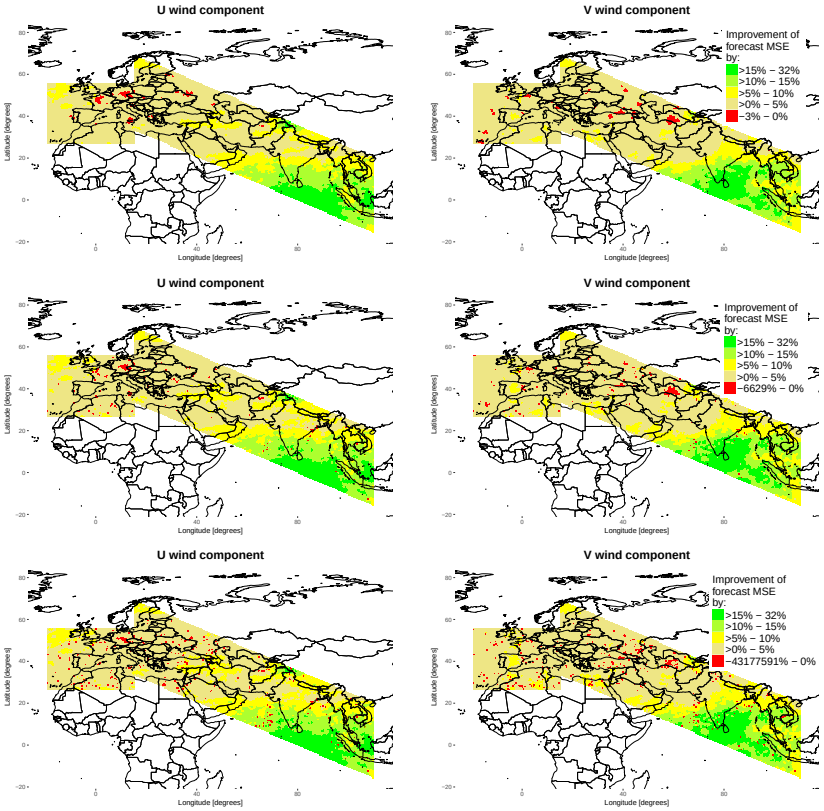


Figure 6.5.: MSE test results for the 5th (top), 10th (center) and 15th (bottom) degree polynomial regression, applied on 24 h forecast data at 200 mbar altitude, for U and V wind components.

The occurrence of scattered negative improvements in areas that the 5th and linear regression performed well in can be explained once more with the prediction error. As the MSE punishes large errors heavily due to the squaring of the error, only a few large errors are sufficient to move a point from the well-improved category to the least.

Results and discussion of higher-order polynomials throughout all time steps The results for all higher-order polynomial regressions are illustrated in figs. A.2, A.3

and A.4. Similar to the 24 h time step are the coverage of improvement patterns. Scattering of single coordinates of deteriorating prediction power are once more prevalent in 10th and 15th order regressions. Maximum improvement steadily declines with a decrease in forecast lead time, as is the case with linear regression in fig. A.1. Negative improvements, contrasting the linear regression's results, however change throughout the different time steps. The greatest deterioration of prediction performance can be found with the 24 h time step.
The number of coordinates falling into the five improvement categories is illustrated in fig. A.9. Trends for all three algorithms are consistent: the best-performing two categories decrease in size with a decreasing forecast lead time, while the middle category retains its number of points. Notably different are the number of deteriorated points for the U and the V wind components. While the number of deteriorated points remains constant for the U wind component, this number grows with the V wind component, especially in the case of the 00 h time step. This observation points to an algorithmic consistency for polynomial regressions, as this behavior is consistent throughout all time steps, for both wind components.

Results and discussion of Support Vector Machine (SVM) SVM results for the 24 h time step are illustrated in fig. 6.6. As with prior algorithms, regions throughout southern and South East Asia report high forecast improvements. These are yet greater with a maximum of 36% improvement than the peaks of all regressions. Both wind speed components' results also show lesser prediction performance throughout all areas west of the Central Asian republics, resembling a similar pattern like with the regression results. The U wind component's results once more show a high prediction performance in areas over northern Pakistan. From this observation the assumption can be drawn that data from this region shows algorithmic prediction robustness, pointing to a reliable and desirable algorithmic performance in this area.
On the other end of the scale, negative improvement numbers differ from those of both the linear and 5th order regression, with higher deterioration. A reason for this may be the SVM's inherent weakness in handling irrelevant input variables (see table 2.2), whereas regressions generally handle this well.

Results and discussion of SVM throughout all time steps Like the results from regressions, a separation exists between areas with higher improvements in Asia and lesser/no improvements in the Middle East and Europe. Also, the maximum value of improvement declines with a decreasing forecast lead time, as well as a increase in the deterioration of results to a maximum of −49% for the 00 h time step. These results point to an ineffectiveness of using the SVM on data from areas roughly

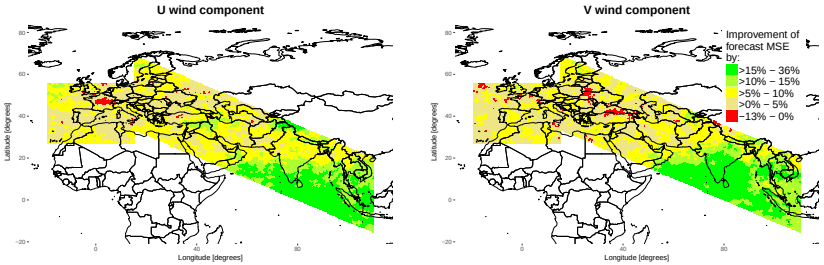


Figure 6.6.: MSE test results for the testing of a SVM, applied on data at 200 mbar altitude, for U and V wind components.

west of Central Asia, especially for short lead times. When inspecting the number of coordinates in the five categories, the results illustrated in fig. 6.7 are found. These results further strengthen the general argument that an SVM is unsuited

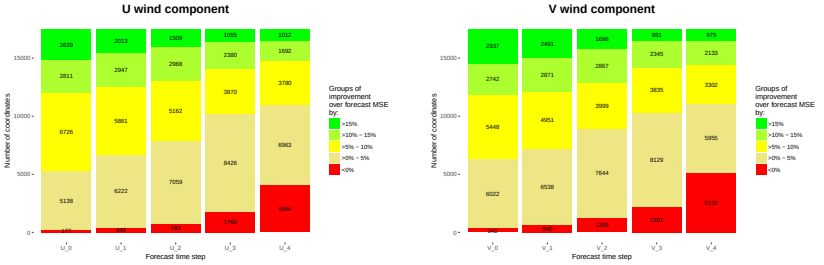


Figure 6.7.: Number of coordinates in each of the five improvement categories for a testing of a SVM on both wind components throughout all time steps, at 200 mbar altitude. Forecast lead times range from 24 h (leftmost column) to 00 h (rightmost column).

to perform predictions on short forecast lead times, as the number of points of negative improvements rise sharply in both wind components between the 06 h to 00 h time step. In both wind components, this growth of negative improvement points can roughly be described as a doubling with every time step. As mentioned above, the reason for this weak prediction performance may be the SVM's inability to handle irrelevant input variables, which may be either the values on geopotential height and/or temperature.

Results and discussion of a Decision Tree and Boosting algorithm Results from these two algorithms are illustrated in fig. 6.8, which differ starkly from any prior results. While a certain degree of consistency in the prediction results can be observed for all prior algorithms, this does not apply for the decision tree nor for Boosting. The latter does not feature a single coordinate location in which its MSE is lower than the original forecast's. Except for an area south of N10°, the decision tree algorithm performs poorly with up to −208% improvement. In the areas in which a positive improvement is observed, positive percentage improvement is similar in range to all prior algorithms. Decision trees and hence, forests,

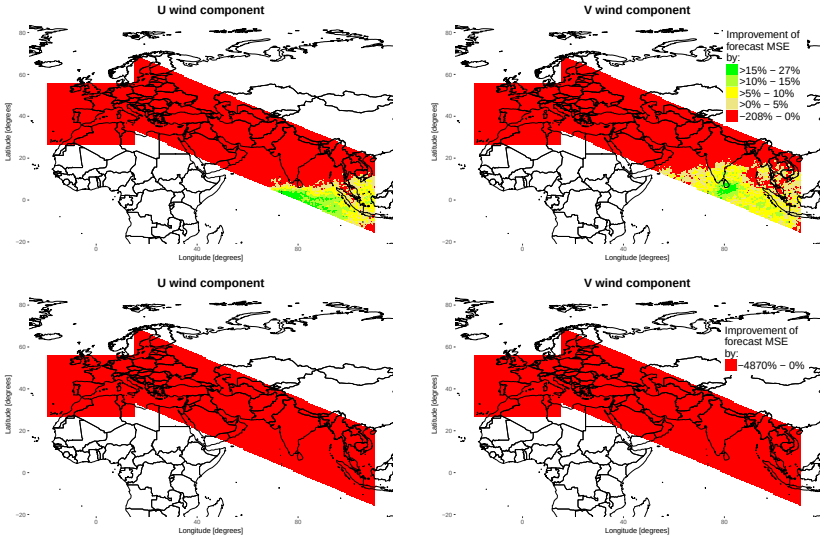


Figure 6.8.: MSE test results for the testing of a Decision Tree (top row) and Boosting algorithm (bottom row), applied on data at 200 mbar altitude, for U and V wind components.

generally perform poorly when extracting linear combinations of data features, as pointed out in table 2.2. The high prediction performance recorded with the linear regression indicates an underlying linear distribution, which in turn explains the poor results recorded for decision trees and boosting.

Results and discussion of a Decision Tree and Boosting algorithm throughout all time steps Results from all other time steps further strengthen the argument that both decision trees and Boosting algorithms are not suited to generate predictions in

this task setting. While the Boosting algorithm does not feature any coordinates with a lower MSE than that of the original data, the decision tree once more features an area south of India in which a limited number of points are located that meet this criteria. When looking at the temporal distribution in fig. 6.9, this trend

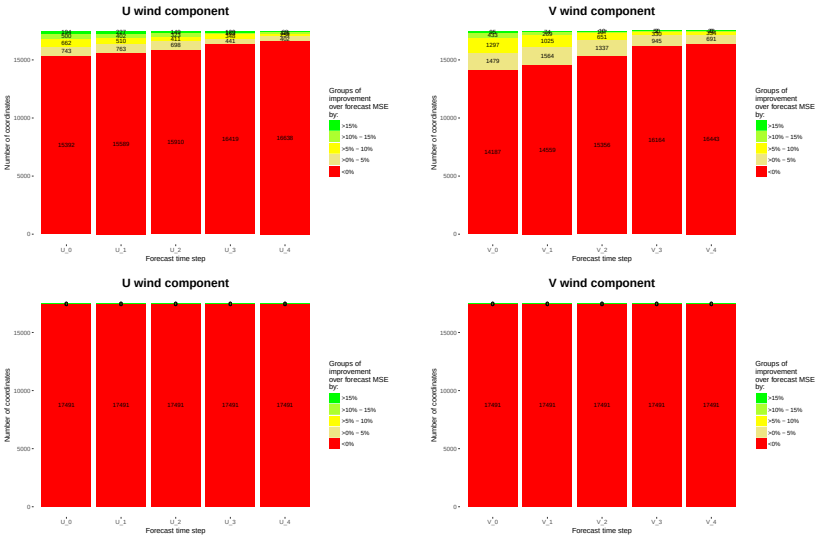


Figure 6.9.: Number of coordinates in each of the five improvement categories for a testing of a Decision Tree (top row) and Boosting algorithm (bottom row) on both wind components throughout all time steps, at 200 mbar altitude. Forecast lead times range from 24 h (leftmost column) to 00 h (rightmost column).

is further confirmed. Looking at decision trees, locations with greater MSEs vastly outnumber any other category. This number further grows with a decrease in forecast lead time.

Results and discussion of a k-Nearest-Neighbors (kNN) algorithm Results for a kNN algorithm on 24 h data is illustrated in fig. 6.10. Several aspects stand out which do not feature in such a way in the results of any other algorithm. Areas with a negative improvement fall into a pattern that includes central to northern Europe. In general, areas above continental Europe feature only areas with minor to negative improvement. This trend changes to high improvement values in areas east of the Caspian Sea until South East Asia, with a maximum improvement of

up to 49% MSE improvement. On the other end, prediction performance declines by up to 25% of the original MSE value. These findings indicate a generally high prediction power, as indicated by literature sources (see table 2.2). This argument cannot be stated for locations in Europe. There, a modification of the k number of neighbors selected may generate better results with this algorithm.

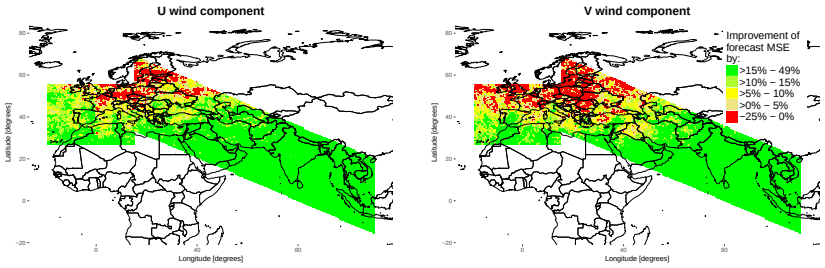


Figure 6.10.: MSE test results for the testing of a kNN algorithm, applied on data at 200 mbar altitude, for U and V wind components.

Results and discussion of a kNN algorithm throughout all time steps The trend identified above of a greater number of coordinate locations featuring the greatest improvements is clearly illustrated in fig. 6.11. Locations with $> 15\%$ improvement over the original MSE dwarf the other categories. The category for negative improvement does grow continuously until reaching about the half with both wind components. While the middle three categories further shrink with decreasing forecast lead time, the best-performing category nevertheless holds a third of locations for the 00 h time step. Such a large share of locations cannot be achieved by any other algorithm, indicating that for shorter forecast lead times, the kNN algorithm may prove to be beneficial. The results throughout all time steps in fig. A.8 reflect the aggregated counts in fig. 6.11. Throughout both wind components, negative improvement locations grow steadily from over continental Europe to cover the area up until the Indian subcontinent and South East Asia. Notable is the strong contrast between the number of negative and most positive locations, as considerably fewer locations exist for the categories in between.

Summary MSE results have shown significantly varying prediction performance behavior. On one end, regressions generally show good performance, with low numbers of locations in which the MSE is greater than the original. Notable is the dominance of the category with $> 0\% - 5\%$ improvement over forecast MSE for regressions and the SVM. Categories are more evenly balanced (see fig. 6.7) than

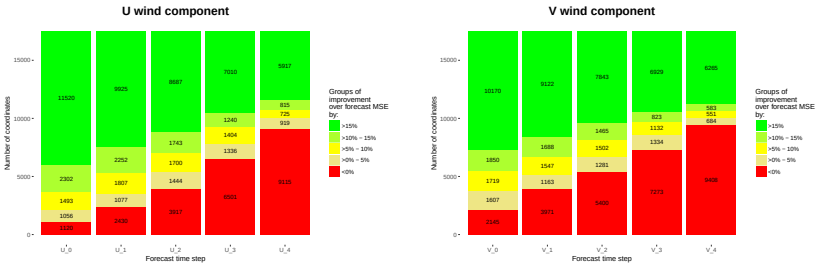


Figure 6.11.: Number of coordinates in each of the five improvement categories for a testing of a kNN algorithm on both wind components throughout all time steps, at 200 mbar altitude. Forecast lead times range from 24 h (leftmost column) to 00 h (rightmost column).

the distribution of the decision tree or boosting (see fig. 6.9) or the kNN (fig. 6.11). Decision trees and boosting feature vastly different results, that show significant disadvantages of utilizing either one of the algorithms for prediction purposes, as the vast majority of locations feature a negative improvement. The kNN algorithm on the other hand features yet different results, as with these the best- and worst-performing categories dominate while the middle categories are dwarfed in the number of locations.

This differing behavior can pose an advantage in the process of selecting the optimal algorithm for an arbitrary forecast. As such, a regression algorithm might be used when it is uncertain whether the algorithm will generate a prediction with an MSE improvement. On the other hand, if it is known that the generated prediction will be more accurate than the original forecast, the kNN algorithm can be applied instead, in the hope of generating a prediction that is yet closer to the true wind value than the regression's prediction.

Seemingly unsuited algorithms for prediction purposes, such as the decision tree and boosting, however should not be written off yet. The MSE which is analyzed at this point is an aggregated result of all test data instances and includes predictions both more and less accurate than the forecast's (albeit with a clear majority of less accurate predictions). If the data instances for which these algorithms generated a desired accuracy can be identified, these algorithms can still be utilized to serve the desired purpose.

Algorithmic robustness

For this subsection, the same vertical level and temporal conditions as in section 6.1.3 are focused on. Robustness in this context is to be regarded as the aggregated results on the performance of the various algorithms on the same test data set. Specifically, this section's focus is on the number of algorithms at an arbitrary location which generate a lower MSE that exhibited by the original forecasts. Also, the type of algorithm with the best and worst performance is of interest and whether any patterns on algorithmic results can be derived from the results.
Fig. 6.12 illustrates the number of algorithms with a lesser MSE than of the original forecasts. In line with the patterns identified in section 6.1.3, the patterns of three major regions can be found. For both wind components, this is the area across southern South East Asia. With the U wind component, this area ends south of India/Sri Lanka, while for the V wind component, this area stretches across parts of southern India. These defined regions exhibit the best results in terms of the number of algorithms which all feature a lesser MSE than the forecasts. Specifically, these are all algorithms except for the Boosting algorithm. The second region

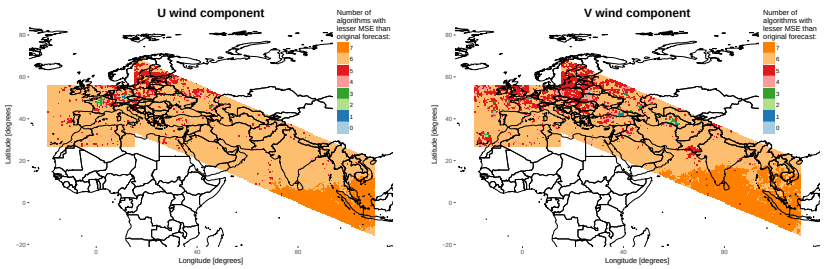


Figure 6.12.: Number of algorithms with a lesser MSE than of the original forecasts. Illustrated are U and V wind components for the 24 h forecast step, at 200 mbar altitude.

is profoundly similar between both wind components, as both feature a vast majority of locations with 6 algorithms performing better on average than the original forecasts. Geographical coverage too is similar, as the region covers all points except the aforementioned locations in Asia as well as central and northern Europe. Throughout this region, a number of pockets of lesser number of algorithms exist. For the U wind component, an exemplary pocket is the area across Sicily, for which up to only two algorithms perform better than the original forecasts. Exemplary pockets for the V wind component can be found over central Turkmenistan and

west of Georgia, which similarly feature up to only two algorithms.
The third area is across central and northern Europe, which shows a scattering of different numbers of algorithms. The plurality of locations in this region feature five algorithms as generating better MSEs than the original forecasts. This observation is valid for both wind components.

Discussion Of the three identified macroscopic regions, the one across South East Asia clearly resembles the results obtained for each algorithm in section 6.1.3. All algorithms with the exception of Boosting performed best in this area. The difference to the major area featuring 6 algorithms is merely the decision tree algorithm. As illustrated in fig. 6.8, it is the only one that only features positive improvements in areas south of about 10° North. The maximum of 7 algorithms can only be found in this region.
With the exclusion of the area covering central and northern Europe, a general finding can be drawn. The patterns identifiable (in terms of number of algorithms) are, with the exception of a small number of pockets, clearly distinguishable from each other. Hence, the central region with 6 algorithms is clearly bounded by a line at about 10° North (for the U wind component) and 15° N for the V wind component. This result indicates that in coherent areas, reliable MSE results are generated. In turn, this proves that the algorithms' MSE results are not a product of chance, as the geographic patterns are this clearly-cut.

Best and worst performing algorithms Fig. 6.13 illustrates the best and worst performing algorithms, respectively. For the worst performing (bottom row), the vast plurality of locations report the Boosting algorithm to feature the greatest MSE in comparison to that of the original forecast. A number of instances featuring the 15th degree polynomial regression exist, primarily in an area off the western Indian coast and the Bay of Bengal. These two groups of points feature for both wind components. The reason for a low prediction performance of a 15th degree regression is stated above in section 6.1: the degree of fit overfits the data, leading to major test errors. Further speaking for this explanation are the positive improvement values recorded with the linear regression; hinting that an assumption that the underlying data's distribution is linear is fitting.
The results of the Boosting algorithm, illustrated in fig. 6.8, already indicate that this algorithm on average is not suited for prediction purposes in this context. This observation is not only limited to certain parts of the area processed, contrasting the higher-degree regressions. It can be derived that this algorithm is not able to learn the underlying aspects of the training data set.
For the best performing algorithms, two algorithms in particular can be highlighted; the SVM and the kNN algorithm. These two dominate the area processed,

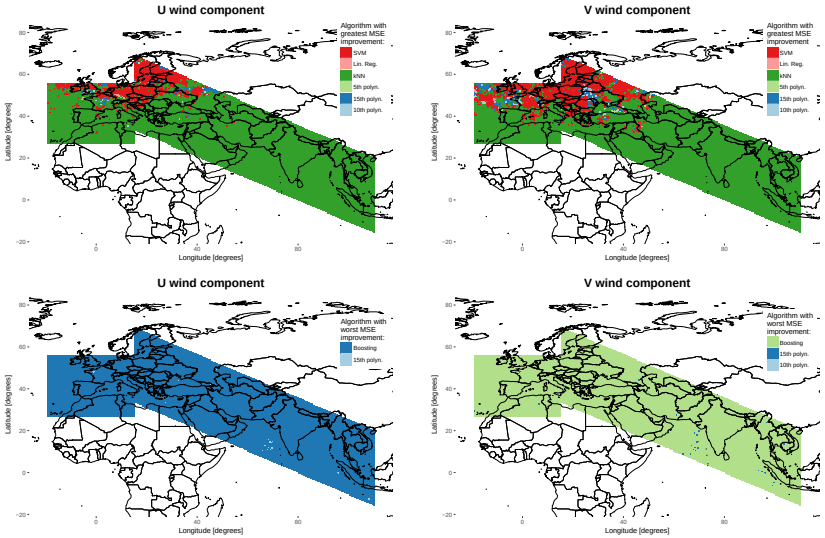


Figure 6.13.: The best (top row) and worst performing algorithms (bottom row). Illustrated are U and V wind components for the 24 h forecast step, at 200 mbar altitude.

with the prior reporting the best predictions in parts of central and throughout northern Europe, for both wind components. The latter algorithm features strongly in almost the entirety of locations in southern Europe, the Middle East and South Asia. Results of the V wind component further show that the third-often best-performing algorithm is the 15th degree regression. This contrasts the observation for the worst-performing algorithm, which also featured the 15th degree polynomial regression. Yet, this result shows that the training of even such a high-degree regression is justifiable and beneficial in certain locations.

Results from other time steps Aggregated results from the remaining time steps are illustrated in figs. A.10, A.11 and A.12. Overall, all patterns identified can also be extended to other time steps. One notable pattern emerging is the increasing number of locations for which the SVM and the 15th degree polynomial regression feature the lowest MSE. This number is growing with a decreasing forecast lead time, as evident in fig. A.11.

Comparison of two test sets In the course of the evaluation of algorithmic prediction power, two further aspects need to be examined. First, this is the question

whether each algorithm has a number of data instances for which only itself and no other algorithm generates an improvement over the original forecast. For this, the Boosting algorithm is a prime example. Mean predictive results do not show a justification for the application of this algorithm overall. If a number of data points exist for which only Boosting generates an improvement, the training of this algorithm is justified. For this purpose, the test results for two separate locations have been examined, namely 50° North 8.5° East and 1.5° North 104° East, both at an altitude of 200 mbar. Fig. 6.14 illustrates the percentages of the data set for which each respective algorithm solely offers an improvement. The first

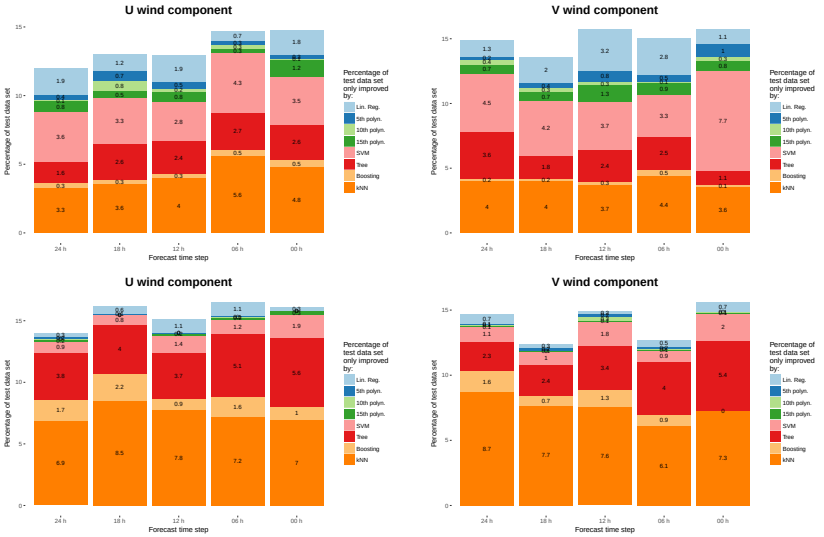


Figure 6.14.: Percent of the total test data set improved only by the respective algorithms. Illustrated are U and V wind components for all forecast steps at 50° North 8.5° East (top row) and 1.5° North 104° East (bottom row), both valid at an altitude of 200 mbar altitude.

location is located near to the city of Frankfurt, Germany. Throughout all forecast time steps, all algorithms exhibit values of $> 0\%$ of the test data set. These range from greater percentages (e.g. 7.7 for the SVM at 00h time step for the V wind component) to the minimum of 0.1% (Boosting at 00 h time step for the V wind component). These results indicate that an algorithm may not perform well on average, yet still generate single favorable predictions. In order to choose the most

beneficial algorithm, another algorithmic layer on top of these must be realized. This derived requirement is the motivation behind the herein described and realized algorithm selection method, as explained in section 5.5.
The second point, represented by the bottom row in fig. 6.14 is a location over Singapore. Results differ from those of the prior location in that the regressions' percentages of the test data set are less than those of the first location. Also, the 10th degree regression features no data instances solely improved for time steps 18 h and 06 h for the V wind component. At the 00 h time step (V wind), the Boosting algorithm also features zero data instances. These results further underline the need for an algorithm selection process, as these algorithms may not produce desired results. Therefore, a means to avoid these in the respective circumstances needs to be realized.
Another factor to be considered for the algorithm selection process is the selection of not one beneficial algorithm, but the best algorithm for an arbitrary data point. On top of this, the selection process also needs to not apply an algorithm should it not improve the forecast's accuracy. This requirement is derived from the results of both location's test results, as illustrated in fig. 6.15. These results show a slight increase in non-improved instances for test results of Frankfurt for lesser forecast lead times. Results for Singapore in turn do not show such an increase, but rather a quasi-constant behavior. Overall, these results suggest that the training of different algorithms is beneficial, as the number of instances improved with at least one algorithm grows to well over 75% in all illustrated conditions. This conclusion yet again demands an algorithm selection process, as the algorithm yielding the greatest improvement should ideally be selected.
The above results have shown the need for an algorithm selection method, the logic of which is presented in section 5.5. For the validation of this method, a separate validation data set is employed. The results of the validation are presented and discussed in section 6.2.

Results from remaining vertical levels

In this section, results from three other vertical pressure levels are presented. These are isobaric levels at 150, 250 and 300 mbar. All training and testing is performed in the same way as all processing performed for the 200 mbar level. Presented herein are the improvement results of all algorithms' MSEs. Identified trends are compared to those found on 200 mbar. A trained linear regression's mean test results are illustrated in fig. 6.16. As with prior observations, a distinctive region of high improvement is found across South East Asia and parts of India, across all pressure levels. A notable difference between the levels can be found in

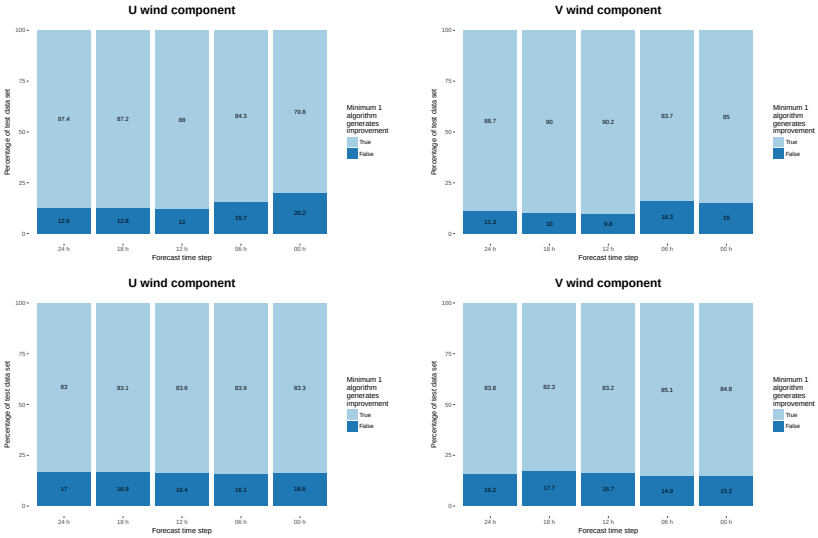


Figure 6.15.: Percent of the total test data set improved by at least one algorithm. Illustrated are U and V wind components for all forecast steps at 50° North 8.5° East (top row) and 1.5° North 104° East (bottom row), both valid at an altitude of 200 mbar altitude.

the number of negative improvement patches, which feature in greater numbers in lower levels, while decreasing at 150 mbar. Prediction performance throughout all levels remain constant for the lower boundary, at $-1/-2\%$ improvement. The positive improvement boundary decreases to a maximum of 24% for the highest level, while remaining at a constant 32% for 250 and 300 mbar.

In summary, these findings indicate that the linear regression generates consistent results, as would be expected since these pressure levels border each other. The increase in negative improvement areas in lower levels might be attributed to greater fluctuations in the data, hence resulting in greater variance. The latter in turn results in a decrease in prediction power for a linear regression.

Patterns found with the other algorithms mimic those found in the prior section. High-degree polynomial regressions generally perform less stable the higher the degree, with the symptomatic spots of major deterioration in prediction performance in areas generally exhibiting good performance. Fig. A.15 illustrates this particularly well.

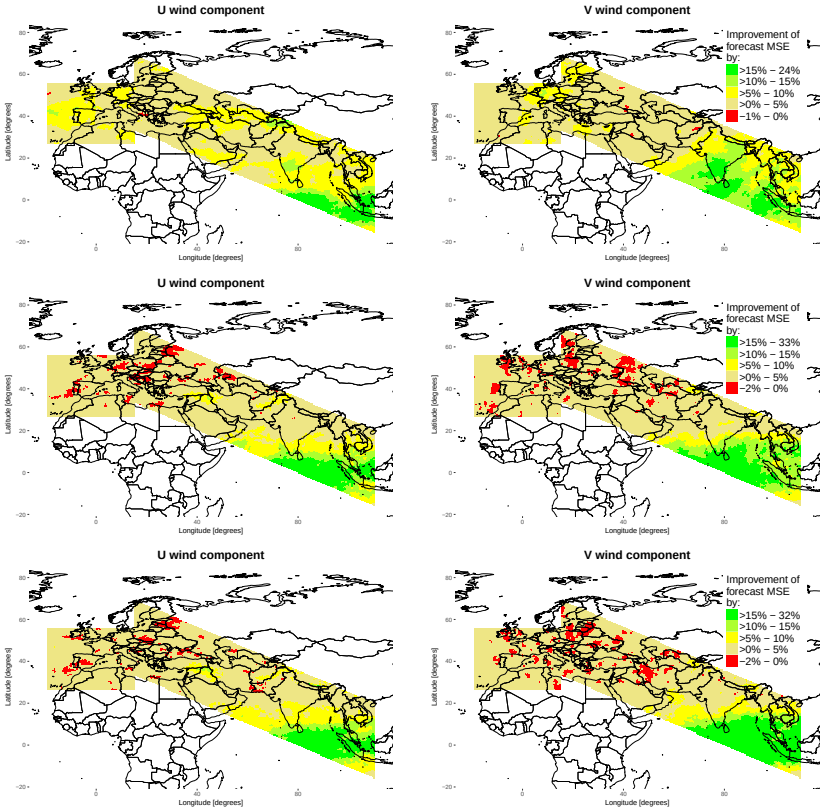


Figure 6.16.: MSE test results for the linear regression for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

SVM and decision trees also show similar results as those from 200 mbar altitude. Results from decision trees especially indicate the algorithm's ineffectiveness in areas outside the aforementioned narrow corridor in South East Asia/India. This trend of weak performance by decision trees is further underlined by the results from the Boosting algorithm. While the results of which show vast locations of weak prediction performance, a few coordinates exhibit positive prediction performance for the V wind component on 300 mbar altitude.

The kNN algorithm once more shows similar trends and patterns as the ones found on 200 mbar. Again, the area above central and northern Europe exhibits

weak, while the rest of the area processed generally shows good prediction performance. The patch of weak prediction performance however decreases in size the lesser the altitude. This finding indicates lower fluctuations in the data, mirroring the indications found with the linear regression.

6.1.6. Hypotheses

This section revisits the hypotheses stated in 6.1.1 and determines whether the above findings result in a falsification or no rejection of these.

Hypothesis H1: *If trained Machine Learning algorithms are used to generate wind speed predictions, then coherent geographic patterns of best- and worst-performing algorithms are retrieved.*

Results showing each location's best- and worst-performing algorithm are illustrated in fig. 6.13. The bottom figures showing the worst-performing algorithms clearly indicate a coherent pattern, with a vast majority of points reporting the Boosting algorithm. Results for the best-performing algorithm too show clear patterns, with the SVM performing the best in northern Europe and the kNN in almost all other areas. The V wind component especially features a number of smaller patches of points reporting linear or higher-degree regressions. Yet, even these exhibit coherent geographical patterns.

Therefore, hypothesis H1 cannot be rejected.

Hypothesis H2: *If trained Machine Learning algorithms are used to generate wind speed predictions, then at least one algorithm's MSE will be lower than the original MSE exhibited by the original forecast at every location.*

The results of the number of algorithms with a lesser MSE than the original forecast, illustrated in fig. 6.12, show that an overwhelming majority of locations feature at least one algorithm. Only two coordinate locations feature zero algorithms with the U wind component, while the V component features four number of locations meeting this category. Promisingly, over 95% of locations feature five or more algorithms which generate a lower MSE than that of the forecasts.

Nevertheless, while vastly outnumbered, coordinate locations where no algorithms generate lower MSE results do exist. Therefore, hypothesis H2 is falsified.

Hypothesis H3: *Each algorithm's MSE will increase with decreasing forecast lead time, irrelevant of the location.*

Discussion of each algorithm's MSE results (see also section A.1) have shown that, while each algorithm's prediction performance varies, all do exhibit a trend to ei-

ther (a) increase the number of coordinate locations with negative improvement values, (b) increase the value of the negative improvement category or (c) both. The reason for this is the decreasing forecast error, as forecasts are generally more accurate the closer in time they are to the event.
Therefore, hypothesis H3 cannot be rejected.

6.2. Validation set evaluation of trained algorithms and the algorithm selection method

This section presents and discusses the results from the evaluation of the trained algorithms and the algorithm selection process, by reliance on a validation data set. First, a number of hypotheses are proposed, after which the results are discussed, leading to a falsification or verification of these.
This evaluation section is divided into four parts. As with the prior evaluation of test results, the focus is first set on a pressure level of 200 mbar, for 24 hour forecast data on both wind components. These results and the corresponding discussion is presented in section 6.2.2. Following this, the focus is then drawn onto the prediction performance throughout the other time steps and patterns retrieved throughout the temporal dimension. The discussion is presented in section 6.2.3. A third evaluation, in section 6.2.4 then divides the results into seasons and discusses patterns retrieved in each separately. The final part of the evaluation will focus on remaining pressure levels and any differences observed in these, as compared to 200 mbar (see section 6.2.5).

6.2.1. Hypotheses

For the evaluation of the algorithm selection method, the following hypotheses have been proposed:

1. **Hypothesis H4:** If Machine Learning algorithms with an algorithm selection process based on historical data are used to generate wind speed predictions, then the MSE results of locations in South and South East Asia will feature greater improvement than of locations in Europe.

2. **Hypothesis H5:** If ML algorithms with an algorithm selection process are used to generate wind speed predictions on a data set spanning a year, then the resulting patterns' MSE will vary depending on the seasons.

3. **Hypothesis H6:** If ML algorithms with an algorithm selection process are used to generate wind speed predictions on data sets of different forecast time, then the resulting MSE will increase with a decrease in forecast lead time.

6.2.2. Evaluation of algorithmic MSE

Like in the evaluation of test results (see section 6.1.5), the predictions generated from the algorithm selection method are compared against the original forecast data, which is assumed to be the closest to the actual wind conditions. Results for the selected area, as depicted in fig. 6.2 for 24-hourly forecasts on a pressure level of 200 mbar, are illustrated in fig. 6.17.

The scale of results reaches from a maximum of 18% improvement of MSE over the forecast data to a deterioration of up to 335% in accuracy. Levels in between these bounds are colored accordingly in the legend of fig. 6.17. Dominating the vast majority of locations is the level corresponding to a maximum of 50% in forecast accuracy; colored yellow. Greater levels of accuracy deterioration, i.e. $> 100\%$, are scattered in smaller pockets solely north of 17° northern latitude. Locations with greatest accuracy deterioration are primarily scattered over continental Europe. Noteworthy in addition is the presence of a greater coherent region of MSE change between -100% to -50% for the U wind component, centered over northern Europe in particular.

Areas of improvement only feature in areas in South and South East Asia. While those of the U wind component feature a greater area south of India and Sri Lanka, locations with accuracy improvement for the other wind component are present along the east and west coast of India. A few locations are also present over the slopes of the Himalayan mountains. The MSE results illustrated in fig. 6.17 share a number of similarities with previous results of single algorithms' performance. In particular, the results for the linear regression serve as a good comparison (see fig. 6.3). In the regression's results as in the current ones, four patterns can be identified. Firstly, the locations with greatest improvement are found south of approx. 17° northern latitude. Reason for this observation may lie in the fact that these areas are prone to monsoon seasons. The volatility of this weather is more difficult to predict, baseline forecast of these regions thereby inhibit greater inaccuracies than those of other world regions. Patterns with lesser accuracy improvement, as well as locations with the greatest deterioration, are only found in areas north of the mentioned latitude. A fourth parallel is both methods' performance along the prime meridian differ to its west and east respectively. This observation hints at a boundary condition inaccuracy, assuming the prime meridian is one such during

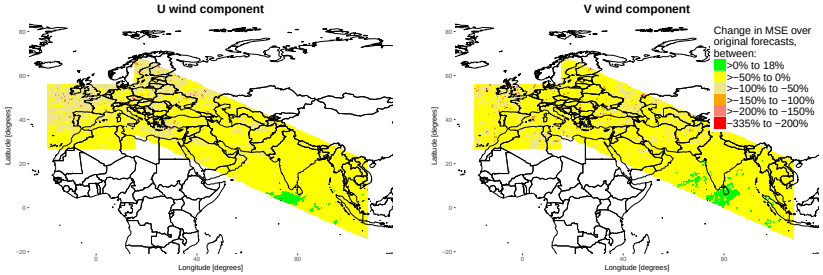


Figure 6.17.: The algorithm selection method's MSE performance results, for data at 200 mbar altitude, for 24 hour U and V wind components.

numerical simulations.
These parallels show that an algorithm choosing from a set of algorithms does exhibit a similar lateral prediction performance and seems susceptible to inaccuracies in the same world regions. A marked difference is yet identified in the magnitude of accuracy deterioration. In results of the linear regression, the most negative results range to −2%, whereas those of the algorithm selection method range up to −335%. This discrepancy points to two possible reasons: a not ideal preset k parameter for the number of neighbors to consider and the heavier punishment of outliers when calculating the MSE.
The challenge to select the most beneficial number of neighbors can be met by repeating the calculations for all k's in question. This is performed in the course of a sensitivity analysis, the results of which are presented in chapter 6.3. In order to discuss the second reason, the percentage of the data set categorically improved in accuracy is plotted in fig. 6.18. These results show that in some locations, only 20% of the data set is actually improved with the current k setting in the algorithm selection method. The upper limit is shown to be at 50% of the data set. Although not the majority, this result nevertheless underlines that the method in principle is able to produce predictions with an improvement over the original forecast. As this data set covers an entire year, it is of interest whether the prediction power of the algorithm selection method varies throughout the year. These results are presented and discussed in section 6.2.5.

6.2.3. Prediction performance throughout all time steps

A number of notable patterns can be identified from the results of the algorithm selection method, executed on data of all available time steps. Judging by the color

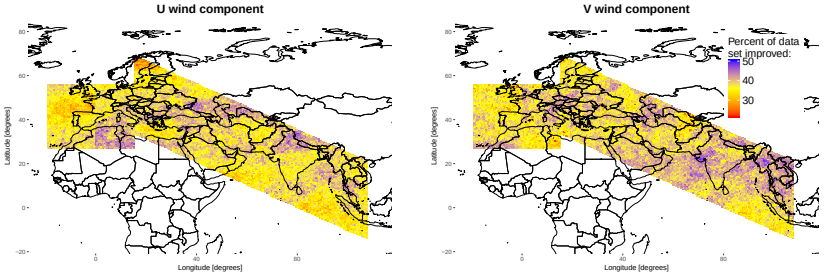


Figure 6.18.: Percentage of data set improved in accuracy, at 200 mbar altitude, for 24 hour U and V wind component forecasts.

coding of the figures in 6.19 alone, a pattern consisting of quantitative differences throughout the time steps can be observed. Results obtained on the 18 hour data show that throughout all locations, the methodology is not able to achieve positive results. This observation is repeated for the 06 hour time step, with results for both wind components exhibiting both qualitative and quantitative similarities. Time steps 12 and 00 (and 24 for that matter, see fig. 6.17) do not show these results. These differences can be identified as a pattern stretching throughout the temporal dimension. By observing these two groups in further detail, a number of other patterns can be retrieved. For the group consisting of 18 and 06 hour forecasts, the patterns' spread across all locations presents a number of similarities. Patches of locations exhibiting the worst performance (in red) can be found throughout Europe and northern Africa, for both wind components. A notable pattern for the U wind results however is a large coherent area over South Asia, particularly the Indian subcontinent. Results for the V wind component in turn show a more scattered pattern of worst-performing locations, throughout the entire focal area.

The other group consisting of time steps 24, 12 and 00 hours, repeats a pattern in which the best-performing locations are found in areas closer to South Asia. These three time steps are also the only ones actually harboring locations with an actual MSE improvement over the original forecasts. Additionally, from the U wind component results of time steps 12 and 00, a boundary already identified in the discussion of test results (see section 6.1.5) of 17° northern latitude.

The latter observation once more points to a consistence in the data: even though the algorithm selection method being another logical layer on top of the trained algorithms, the pattern of better performance is found in South Asia, also observed in the 24 hour results (fig. 6.17) and throughout the test results of section 6.1.5. Due

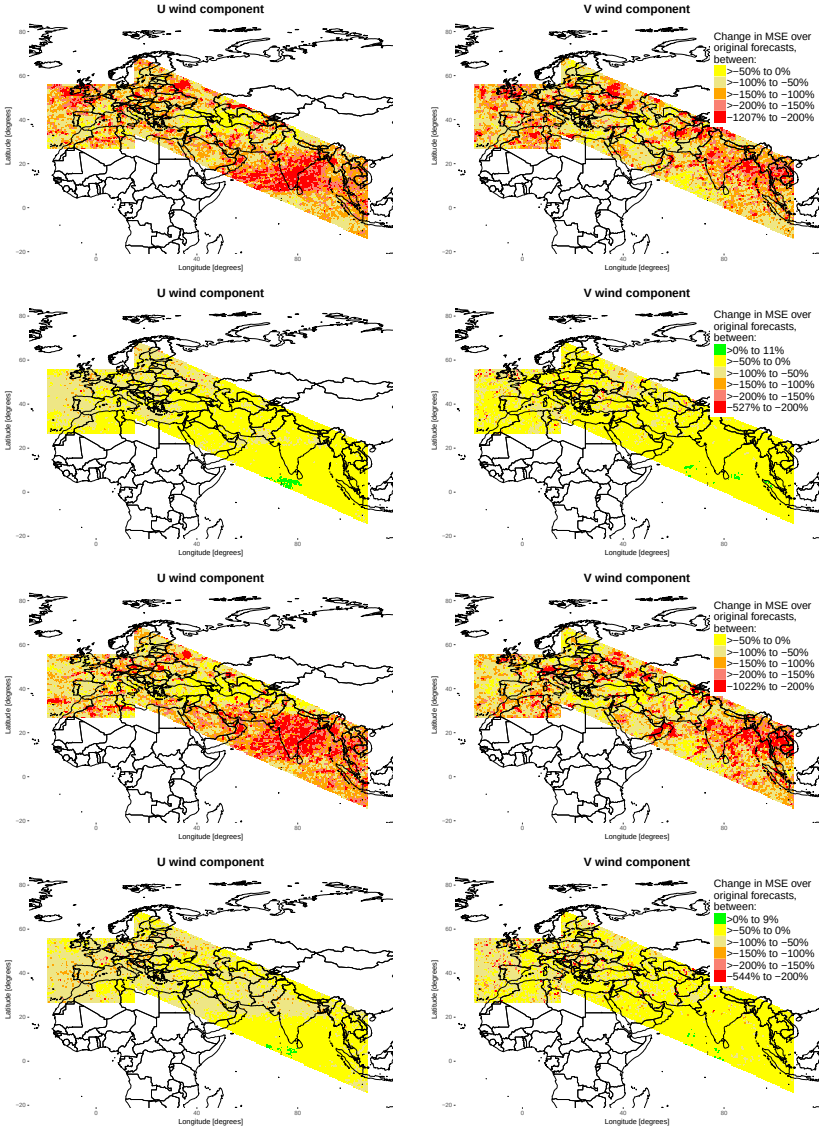


Figure 6.19.: MSE performance results of the algorithm selection method for 18 (top) to 00 (bottom) hour time steps in 6-hourly intervals, at 200 mbar altitude, for U and V wind component forecasts.

to the proximity to regions affected regularly by monsoons, this is once more hypothesized to be the root cause of this phenomenon.
Less conclusions can be drawn to explain the major differences in quantitative results between respective time steps. Similar regional patterns exhibited in 18 and 06 hour time steps point to a systematic phenomenon. As the logic applied to all data is the same, the only difference is the data itself. The conclusion to this observation is that the underlying data bears patterns that are not suited to work effectively with the algorithm selection method in this current configuration. However, a different k number of neighbors to be considered may provide a better result for these time steps. Likewise, the green areas in time steps 24, 12 and 00 hours where a positive result is recorded, may cover greater areas than currently observed. To investigate these hypotheses, a sensitivity analysis is performed in chapter 6.3.
Investigating the MSE alone in this application is not sufficient, as the number of instances actually improved is of interest. Fig. A.24 illustrates the percentage of the respective data set categorically improved. For the two time steps with overly negative changes in MSE, 18 and 06 hours, the results show that in the worst cases, 10% of the data set is categorically improved over the original forecasts. The maximum percentage number is set at 40%. In contrast, the other group of time steps peak at 50%, agreeing with the general, less negative trend identified in fig. 6.19. Even this display of categorically improved data instances nevertheless consider the data set stretching over an entire year of data. It is of interest how parts of this data set perform on their own. These parts herein are defined as seasons, with the goal being to identify seasonal dependencies across the entire area of concern. Section 6.2.4 elaborates on this question.

6.2.4. Prediction performance by season

The prior two sections presented a view across the entire data set. A more differentiated view of the single seasons' results can present more insight into underlying patterns. Also, if a certain season or temporal period is determined to work very undesirable, this part could be left out of any further usage scenario of this method. The seasons are defined in the following:

- Spring: March, April, May
- Summer: June, July, August
- Autumn: September, October, November
- Winter: December, January, February

The data set is respectively filtered for these seasons, with the results for the seasons plotted in fig. A.23. The results of both wind components show two general patterns: results excluding South Asia and South East Asia are qualitatively similar, whereas these areas present the only positive results of the method. Coverage of the latter pattern varies significantly throughout the seasons. Judging by the number of partitions/locations, the method is least effective in winter and most in summer. This holds true for both wind components. By comparing the mean results across the entire year (see fig. 6.17) against these seasonal ones, it can be shown that the most positive and negative peaks vary by close to 100%. The greatest difference is recorded between the yearly and spring results, in which the most negative values are set at -335% and -701%, respectively. Likewise, this difference can also be recorded for the most positive peak.
Prior discussions herein (see sections 6.1.5 and 6.2.3) have drawn the conclusion that positive results of the algorithm selection method as well as algorithmic test results are connected to the seasonal monsoon in South Asia. This hypothesis can be confirmed by the summer results illustrated in fig. A.23. Monsoon season in India typically lasts from June/July until at least September [146], the meteorological processes being very complex and hence more difficult to predict than during other seasons. This forecast inaccuracy is then picked up by the method herein, with the time frame set for summer being defined to be June through August. Results for spring, during which the monsoon has not yet begun, show a lesser number of positive changes. Autumn, which covers September (the last monsoon month) shows a decrease in the method's effectiveness, albeit greater effectiveness than during winter.
These insights show that a pattern of seasonal dependency can be observed throughout the entire region of focus. This dependency is seemingly interpretable with meteorological phenomena, such as the monsoon in South Asia. Generally, it is of interest whether positive (as compared to the original forecasts) results, as well as those less accurate, in turn generate a likewise effect on the performance of the flight planning process.

6.2.5. Prediction performance throughout other pressure levels

Fig. A.21 illustrates MSE performance results across all four pressure levels. Similar regional patterns are observable throughout these altitudes. For the V wind component, the number of locations with positive improvement grows with a decrease in pressure level. This observation indicates that the method works for other pressure levels and may even feature more positive results on lower ones, given the

trend for the V wind component.
By plotting the results of the 300 mbar pressure level across the four seasons, similar patterns for summer and autumn can be observed across South Asia. For both wind components, these seasons generate the greatest coverage. This is a result which agrees with the hypothesis in section 6.2.4, which connects this region of increased prediction performance with the regional monsoon season.

6.2.6. Processing times for the algorithm selection method

The process of selecting the optimal algorithm, as detailed in section 5.5, is performed using a maximum of 97 CPUs in the data cluster. Additionally, 25 containers are allocated by Spark to support the job. Each task of the total job runs in one container. As the number of completed tasks grow, the number of CPUs and containers steadily decrease. Like the training and testing processing job, the total number of partitions is divided into iterations per longitude. The number of parallel partitions therefore varies per longitude, between 232 and 284. Running in the same bounded area as illustrated in fig. 6.2, processing time per longitude is set at 5 hours. The total required time for all partitions therefore covers just under 54 days. This number however does not include interruptions due to maintenance tasks or other issues.

6.2.7. Hypotheses

This section revisits the hypotheses stated in 6.2.1 and determines whether the above findings result in a verification or falsification of these.

Hypothesis H4: *If Machine Learning algorithms with an algorithm selection process based on historical data are used to generate wind speed predictions, then the MSE results of locations in South and South East Asia will feature greater improvement than of locations in Europe.*
Results shown in fig. 6.17 indicate a pattern difference in prediction performance between South Asia and Europe. This finding is mirrored in the results obtained in the prior evaluation of test results, as discussed in section 6.1.5. It can be concluded that the meteorological phenomena in this region actually lead to the algorithm selection method generating more accurate predictions. This conclusion is further supported by the identification of a large lateral pattern of improvement for data valid in the South Asian monsoon season, as illustrated in fig. A.23.
While these results hint at a verification of this hypothesis, some of the results discussed point to the contrary. Particularly, the results illustrated in fig. 6.19 for

forecast time steps 18 and 06 hours show that locations in South Asia perform worse than in Europe. Therefore, hypothesis H4 is falsified, as a verification cannot be produced in every instance.

Hypothesis H5: *If ML algorithms with an algorithm selection process are used to generate wind speed predictions on a data set spanning a year, then the resulting patterns' MSE will vary depending on the seasons.*
Seasonal results illustrated in fig. A.23 point to a heavy correlation seasonal meteorological phenomena and prediction performance. This is shown the varying number of locations reporting MSE improvements over South Asia. Most prominent is this effect during summer, which consists of three out of four months of monsoon season in this area. This effect can also be observed in a more limited number of locations during autumn months. In turn, during winter and spring, no significant pattern is observable. Therefore, hypothesis H5 cannot be rejected.

Hypothesis H6: *If ML algorithms with an algorithm selection process are used to generate wind speed predictions on data sets of different forecast time, then the resulting MSE will increase with a decrease in forecast lead time.*
Results over decreasing forecast lead time, as illustrated in fig. 6.19, show that a consistent growth in prediction performance does not follow a steady decrease in lead time. Rather, the time steps representing 18 and 06 hour forecast steps show less accurate results than the three other time steps. A trend associated with all single algorithms (see section 6.1.5) over decreasing forecast lead time, cannot be identified for the algorithm selection method. Therefore, hypothesis H6 is falsified.

6.3. Sensitivity analysis

Results from the algorithm selection method of section 6.2 have shown that the method realized is successful in a number of locations. Locations with a positive impact of the methodology prove its theoretical feasibility as a proof of concept. The method is run with a pre-defined constant value of $k = 6$ neighbors. This value may not be beneficial for all locations. The algorithm as outlined in algorithm 4 is run for a range of $k = 1, ..., 20$ neighbors. The algorithmic logic is not modified for all k, with the exception of $k = 1$. For this case, the rules established for two or more neighbors do not apply and are ignored. To shorten processing time, the sensitivity analysis is only performed for two locations on a pressure level of 200 mbar:

- Frankfurt, Germany: N50° E8.5°

- Singapore, Singapore: N1.5° E104°

These two locations have been selected due to their discrepancy as shown in results in chapter 6. The location representing Singapore has consistently shown good prediction performance for both the single algorithms, as well as the algorithm selection method. On the other hand, Frankfurt and locations in Europe in general have shown fair performance for algorithm testing, yet poor performance for the algorithm selection method. These differences in results, as well as the climatological discrepancies may show different trends as to which k number of neighbors is most beneficial for each respective location.
Results are presented and discussed first on a yearly basis. Wind components are treated separately, with the locations' results over k compared against each other. Yearly results are then further refined into seasons.

6.3.1. Yearly results

Yearly results for the U and V wind components are illustrated in figs. 6.20 and 6.21. Two distinctive patterns, clearly separating the results for Frankfurt and Singapore, can be observed. The lowest results for Frankfurt are observed for $k = 1, 2$ and rise steadily up to $k = 10$ before dropping again. Contrary to this trend are results for Singapore, which feature a maximum at $k = 1, 2$. All other k feature more negative results. These identified discrepancies underline the meteorological differences at these two locations. A maximum observed with the least possible number of neighbors, i.e. $k = 1$, indicates that the model for Singapore works best with the lowest possible generalization. The reasoning may be that tropical weather phenomena fluctuate more (than those valid for Frankfurt). Areas of similarity in the d dimensions that the kNN is operating in are therefore smaller. By reducing the number of neighbors to a minimum, the likelihood of leaving this area of similarity is thereby reduced. In essence, areas of similarity in the d-dimensional space are small, best described by a single point. This assumption indicating high fluctuation agrees with the conclusion that in tropical regions, meteorological predictability is generally poor [147].
Contrary to this is the observation that for Frankfurt, the U component maximum is given at $k = 10$ and for V at $k = 20$. This behavior in turn indicates that the kNN is more successful with more neighbors concerned, i.e. with a lower variance. Poor performance with a high variance indicates that the algorithm tends to model noise in the data set. When interpreting this in a meteorological context, this data set learns features more from wider areal patterns, hinting at less fluctuations in the data.

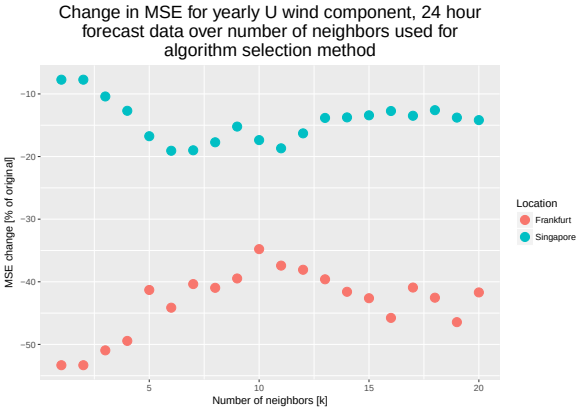


Figure 6.20.: Change in U wind component MSE in % of the original forecasts, generated with the algorithm selection method (see alg. 4) for various $k = 1, ..., 20$. Results are generated with 24 hour forecast data from the entire year, valid at Frankfurt and Singapore.

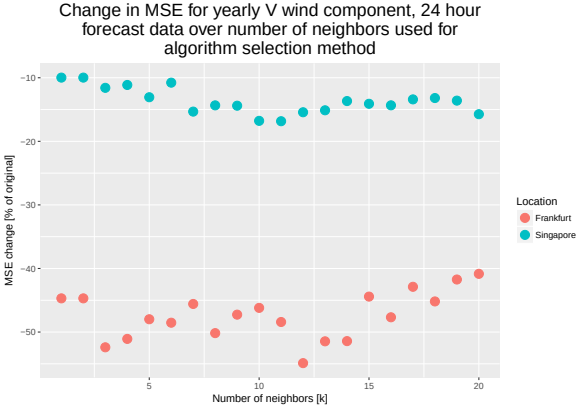


Figure 6.21.: Change in V wind component MSE in % of the original forecasts, generated with the algorithm selection method (see alg. 4) for various $k = 1, ..., 20$. Results are generated with 24 hour forecast data from the entire year, valid at Frankfurt and Singapore.

6.3.2. Seasonal results

Seasonal results in fig. 6.22 for Frankfurt show similar patterns throughout all four seasons. These climb starting at $k = 1$, peak at a range of between 5 to 12 before decreasing again. The fact that this qualitative trend is mirrored throughout all seasons shows that the underlying seasonal effects are similar. This conclusion can only be drawn in part from results concerning Singapore. While both the range and qualitative behavior for $k = 1, ...4$ neighbors are similar for all seasons except summer, results of higher k showcase different trends. For spring, the second-highest MSE change is observed for $k = 18$, indicating that the algorithm selection method works increasingly better with a growing number of neighbors. Such a behavior indicates that the method improves with a decrease in variance. This cannot be said for the method's trend for winter, which does not show a recovery after $k = 9$. Data regarding autumn is an outlier in this regard. As mentioned, the results for the first three k are similar to results for spring and winter. However, the general trend shows no fluctuations greater than 5% and the best result recorded for $k = 13$. This behavior indicates that the number of neighbors considered are not as important as for the other seasons. Such stability is a possibility if the data exhibits consistency. In other terms, this can be regarded as the data being very similar in a wider d-dimensional kNN space. A prime example is the deviation between forecast and re-analysis values. For autumn data, this discrepancy is quantitatively similar than for the other seasons. Summer in particular shows major fluctuations in both locations. The prior-identified trend in section 6.3 of high fluctuations in tropical regions is visible for Singapore results. With no general consistency in the results observable, the reasoning is that even a minor change in the number of neighbors leads to a significant bias. In such volatile data space, as the results indicate, the best solution is to avoid the areal bias and utilize only a single neighbor.

As identified in the yearly MSE results the best results are recorded for $k = 1, 2$. A minimum is recorded for $k = 11$, with an increase in MSE results after this point. This trend for summer is mirrored qualitatively in the yearly results (see fig. 6.20). Reason for this may be that the greatest deviations are recorded in this season, which thus influence the yearly results more than those of other seasons.

In conclusion, it can be shown that the best value of k to utilize varies not only by location, but also by season. The best values cannot however be defined deterministically, but through calculation of all possible scenarios.

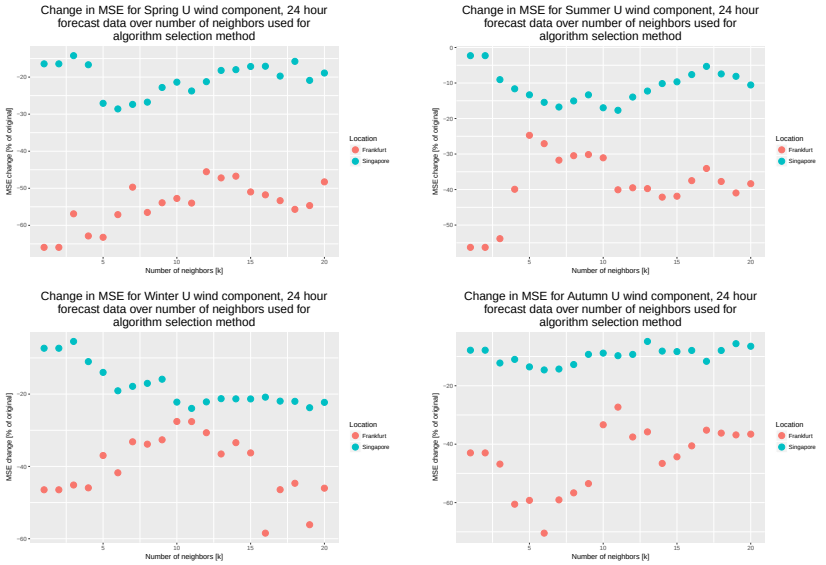


Figure 6.22.: U wind component MSE results for different k of the algorithm selection method across all four seasons, clockwise from spring (top left) to winter (bottom left), for 24 hour forecasts.

6.4. Summary

The results show that some algorithms perform poorly on a vast number of locations, such as Boosting or the decision tree, while others, especially the kNN exhibit high prediction power. Generally, it is shown that a majority of algorithms feature predictions which constitute an improvement over the original forecasts' accuracy. A second algorithmic layer, the algorithm selection method, is developed to utilize historical data to select the presumably most accurate predictive algorithm. This method is evaluated using a validation set spanning one year of data. Results show that this method's predictive power is limited to a number of world regions and correlates with seasonal meteorological phenomena.

The sensitivity analysis performed herein has shown that the algorithm selection method is dependent not only on the location of concern, but also of the season concerned. Moreover, it can be concluded that the most beneficial k number of neighbors for each case needs to be determined by calculating every possible sce-

nario, as a deterministic solution cannot be defined.

7 Concept validation in the context of flight planning

This chapter focuses on the validation of the realized system for uncertainty prediction, as detailed in chapter 5. A testing of algorithmic prediction performance and a validation of the algorithm selection process have been performed. The focus in these two steps is primarily on the prediction of the specific discrepancy between a given forecast and the true wind speed. In this chapter, the focus is on the effect the predicted wind speeds in turn have on the flight planning process. The aim of doing so is to quantify the impact this concept has on performance indicators applicable in flight planning.
In a first step, the idea and scope of this validation is outlined, after which the schematic approach is described. Following this, the focus on a specific performance area in flight planning is explained. Lastly, specific flights are presented, for which this concept is to be validated. Validation results, including hypotheses and discussions are presented in the last part of this chapter.

7.1. Validation idea and scope

Flight planning fundamentally relies on a number of different factors [10]. The quality of this mandatory process in aviation may vary, depending on the availability of information describing these factors, as well as the accuracy of this information. Additionally, the term *quality* is needed to be defined precisely and objectively. One significant factor influencing the flight planning process is weather. Typically, planning engines rely on forecasts of temperature and wind speeds, valid at specific pressure levels.
Prior research, such as by Cole et al. [6] has identified wind forecast errors as the greatest source for trajectory prediction errors. The primary objective in this thesis is to predict a forecast's uncertainty and utilize this knowledge to improve the same forecast's accuracy. It is therefore of interest whether an improved forecast, assuming it is more accurate than the original one, yields a flight plan that is of a greater quality than the original one.
In essence, any flight plan is a prediction of future events. Ideally, the best case sce-

nario is that every time a plan is produced, the corresponding actual flight occurs exactly as planned. In this way, the plan's predictive power, or *predictability*, would be maximal. In reality, the Air Traffic Management (ATM) system is so complex that only a fraction of flights are conducted exactly per their plan. Flight planning influences an airline's scheduling, which is further influenced by other factors, such as crew rostering, maintenance routing and fleet assignment [148]. A decrease in flight plan predictability can therefore translate to scheduling problems, which in turn may cause disruptions and added costs to the airline. The underlying reasoning is therefore that a decrease in forecast accuracy decreases a flight plan's predictability and hence negatively affects scheduling.
The concept realized herein has proven to be able to increase forecast accuracy in limited locations and different forecast lead times, as shown in chapter 6.1.3. This raises the question whether the logic holds true: *Would an increase in forecast accuracy lead to a greater predictability, i.e. lesser discrepancy between the actual route and the flight plan?*

7.2. Schematic approach

The schematic discrepancy between a flight's trajectory and its corresponding flight plan is illustrated in fig. 7.1. Depicted is a the actual trajectory and two further paths indicating two differing flight plans. The x axis represents the actual trajectory, to which the flight plans are compared against. On the y axis, an arbitrary dimension is plotted. In order to quantify the effect of forecasts with greater accuracy on

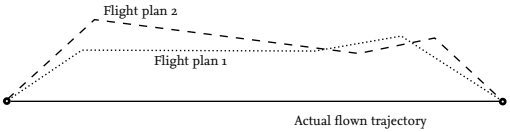


Figure 7.1.: Two different flight plans with their respective discrepancies to the actual flown trajectory.

flight plan predictability, both flight plans are not compared against one another. Rather, the actual trajectory serves as a reference. Both flight plans' discrepancies to the actual trajectory are determined in a first step. Only in a second step are these differences compared. Specifically, this approach is realized by the process illustrated in fig. 7.2. A database holds all forecast data for this validation. This and the algorithm selection process are identical to those illustrated in fig. 6.1. Four different data sets are then extracted directly or indirectly (with the algorithm selection method in between; for details see chapter 5.5) from the validation database.

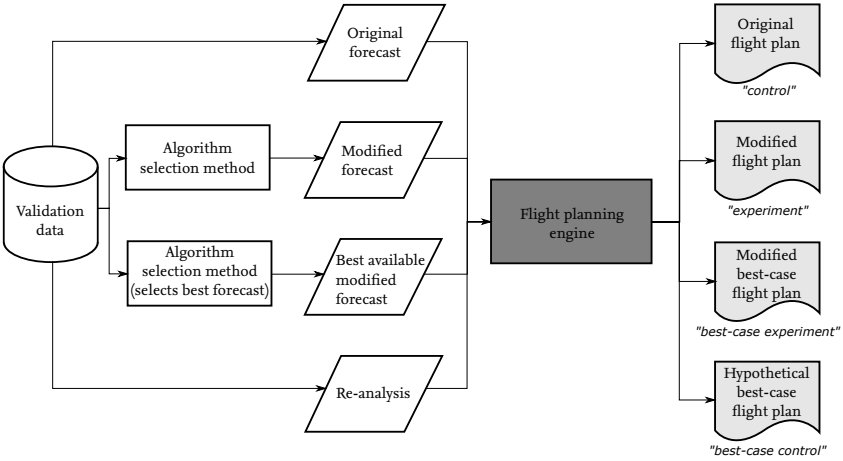


Figure 7.2.: Process for the generation of four different flight plans; one each generated with the original forecast, modified, best-case modified and re-analysis data.

The first set of data is forecast data, which resembles the same data used by the airline for the originally-filed flight plan. By utilizing this data for flight planning, a control/baseline most similar to the airline's original flight plan can be realized. The second flight generated (illustrated in fig. 7.2 as the modified flight plan) is the experiment. This flight plan is the result based on the output of the algorithm selection method. Together with the original flight plan and the actual flight trajectory, the benefit of the concept detailed in chapter 3 can be evaluated.
On top of these two flight plans a further third and fourth are generated. The third flight plan is generated on the same algorithm selection method, albeit with the best algorithm selected at all times. By doing so, an upper boundary of the method's benefit to flight planning can be determined. A final fourth flight plan is generated on the basis of re-analysis data. As the latter represents the most accurate depiction of the state of the atmosphere available [100, 102], it effectively represents the upper-most boundary in terms of efficiency for the flight planning process - since the data is assumed to be the most accurate. The flight planning engine utilized herein is Jeppesen's *FlitePlan Core* product. This engine features the possibility of evaluating Eurocontrol Route Availability Document (RAD) routes and hence provides a greater level of realism for flights traversing European airspace than entirely ignoring these routes.

A creation of the original flight plans is necessary, as these filed by airlines are

not freely available. Making matters yet more difficult is the fact that flight planning settings are confidential pieces of information and thus are not available publicly. By utilizing the same flight plan engine settings for all flight plans, all generated errors will be constant. In this way, the error caused by the fact that the airline's exact flight planning settings have not been used, is negated.

7.3. Performance criteria

Comparison of flight plans with the actual flight trajectory is limited to a number of factors. This is due to only the trajectory (which includes location, time, altitude and ground speed) being known. Other factors, such as actual costs or fuel consumed, are sensitive information and not published by airlines. Any comparison of a given flight plan to actual data is limited to the locational data with a temporal tag. All performance criteria can therefore only be derived from this information reported through Automatic Dependent Surveillance - Broadcast (ADS-B) from aircraft.
The Key Performance Indicator (KPI) of focus, as mentioned above in 7.1, is predictability. As such, a lesser deviation from the actual trajectory, irrelevant of the criterion, is defined as desirable. Given the dimensions present in the trajectory data, the following criteria can be derived directly[1]:

1. **Difference in flight duration:** defined as the time difference between the first data point logged after take off and the first one after touch down.

2. **Difference in flight distance:** defined as the total distance covered between takeoff and landing.

3. **Mean cruise speed:** defined as the mean cruise speed at all logged points between the Top of Climb (TOC) and Top of Descent (TOD).

4. **Accumulated lateral deviation:** defined as the shortest distance between a given point to the actual trajectory.

5. **Accumulated vertical deviation:** defined as the shortest distance between a planned and the actual altitude.

Among these five, a differentiation between the first two and the other three can be drawn. While all of these can serve as a metric expressing predictability in their

[1] A foregone thesis by Haude [149] identified these among a multitude of other possible criteria.

own stated dimension, the first two present a more suited and concise quantification of this KPI. Both flight duration and distance are the two major aspects during planning, as these directly influence the amount of fuel and backup fuel. The more accurate these two criteria can be predicted, the more accurate the amount of needed total fuel is needed to be carried.
Being able to predict the mean cruise speed more accurately prior to take off may be of interest for tactical decision support systems. Logically, this criterion is effectively a variable expressed in part by the flight's duration and distance, besides the take off and landing phases of flight. Due to this, the mean cruise speed does not generate added information on predictability - since it is already part of the function of duration and distance. This argument is also applicable to the criteria of lateral and vertical deviation. As with mean cruise speed, these are of interest for tactical decision support and online trajectory optimization. Again, these deviations once more are a function of flight duration and distance.
Primarily, from an airline's point of view, predictability is desirable to estimate the resources and effort needed to safely and most efficiently conduct a flight. In essence, airlines aim to achieve maximum profit using minimal resources or effort. This implies that the overall costs are of interest. While criteria 3-5 do influence costs, they do so only indirectly, as they are a function of total flight duration and costs. In essence, an airline is primarily concerned as to how far costs can be reduced, principally irrelevant of the route taken from the point of origin to destination.
Due to the above points, the evaluation herein will be limited to the differences in flight duration and distance.

7.4. Flights selected for validation

For the validation in the context of flight planning, three flights have been chosen. Each of these corresponds to a short-, medium- or long-haul flight respectively. The chosen medium- and short-haul flights are illustrated in fig. 7.3: TUIfly flight TUI2148 from Hanover, Germany to Fuerteventura, Spain and Garuda Indonesia flight GIA867 from Bangkok, Thailand to Jakarta, Indonesia. TUI2148 is conducted with a Boeing B737-800 aircraft and is regularly scheduled with a flight time of 4 hours and 30 minutes[2]. Scheduled departure and arrival times are set for 1:35 pm and 6:05 pm Coordinated Universal Time (UTC). The flight is performed from Wednesdays through Saturdays and Mondays.
The short-haul flight GIA867 is also performed with a B737-800 aircraft. Flight

[2] According to website FlightAware (`https://de.flightaware.com/live/flight/TUI2148`); accessed: 20-10-2017.

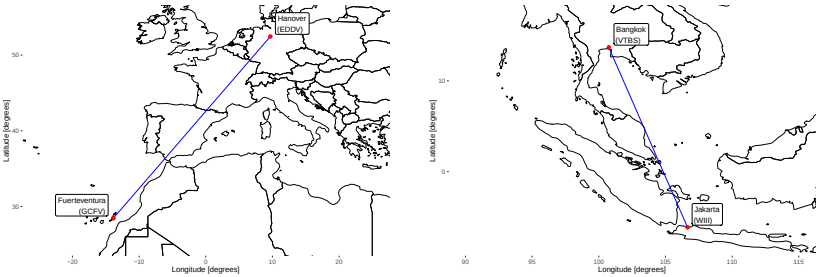


Figure 7.3.: Mid- and short-haul routes considered in this thesis: TUI2148 (left) and GIA867 (right).

Figure 7.4.: Long-haul route considered in this thesis: DLH778.

time is set at 3 hours and 10 minutes, with departure at 7:10 am and arrival at 10:40 am UTC[3]. GIA867 is performed every day of the week.
The long-haul flight selected for validation, Lufthansa DLH778, serves the Frankfurt-Singapore route daily (route illustrated in fig. 7.4). Deployed is an Airbus A380-800 aircraft; scheduled for a total flight time of 12 hours and 2 minutes, with departure at 7:55 pm (during winter months at 8:55 pm) and arrival at 7:57 am (8:57 am) UTC[4].

The geographical distribution of these flights stems from the area of analysis, as illustrated in fig. 6.2. From test results (see chapter 6.1), an area of approximately

[3]According to website FlightAware (`https://de.flightaware.com/live/flight/GIA867`); accessed: 20-10-2017.

[4]According to website FlightAware (`https://de.flightaware.com/live/flight/DLH778`); accessed: 20-10-2017.

south of N17° can be identified, which features maximal prediction performances. The effect of these on the predictability of flight plans is of great interest, as the assumption is that the greatest gains are found in areas with the greatest improvements over original forecasts.
TUI2148 is chosen, as TUIfly GmbH has offered the actual filed flight plans of this flight number for research purposes herein. Additionally, these plans have been generated with the same flight planning engine, FlitePlan Core. Since the flight plans herein are also generated using the same engine, a direct comparison between plans can be justified. Lastly, long-haul flight DLH778 is chosen to cover the broad corridor ranging from Europe to South East Asia.
A list of dates on which these flights were performed is determined for all three flights. The earliest date is July 1st, 2016 through June 30th, 2017. This list is relied upon in the process of data retrieval on flight tracks and weather forecast data, as detailed in section 7.5. Whether or not a date exists on this list is determined by the availability of the flight trajectory in the database.

Required time steps All necessary times of all three flights are detailed in table 7.1. Departure and arrival times serve to determine first the time at which the respective flight's flight plan is created. This lead time to departure is set at five hours prior for GIA867, approximately 11 hours for TUI2148 and 2—3 hours for DLH778. All of these are sensitive airline data and thus not published by these. The time of flight planning is then utilized to determine a reference time, i.e. the next prior 6-hourly time at which weather forecasts are available. For GIA867 and TUI2148, both planned in the early morning hours, this reference is midnight. The long-haul flight is assumed to be planning no earlier than 18:00 pm, which can also be used as the reference time.
These reference times are important for determination of the forecast steps for the flight planning engine. GIA867 departs at 7:10 and arrives at 10:40 am. Forecast data hence has to cover this time, requiring a forecast at 6:00 and 12:00 am. Seen from the reference time, these time tags are forecasts for 6 and 12 hours respectively. With the same logic, the forecast steps for the other two flights are determined.

7.5. Data handling process for flight plan/trajectory comparison

The evaluation process outlined in fig. 7.2 necessitates four unique data sets: the original forecast, modified and best-case modified and re-analysis data. Each serves as weather data input to generate the original and the respective flight plans. This section provides details on the process utilized to retrieve and handle this data, as

Flight no.	Departure time	Arrival time	Flight plan generation time	Reference time	Forecast steps required
GIA867	7:10 am	10:40 am	2:10 am	00:00 am	$+6h$, $+12h$
TUI2148	13:35 pm	18:05 pm	2:30 am	00:00 am	$+12h$, $+18h$
DLH778	19:55 / 20:55 pm	7:57 / 8:57 am	18:00 pm	18:00 pm	$+0h$, $+6h$, $+12h$

Table 7.1.: Summary of flight times, including assumed flight plan generation times and derived forecast steps required. All times in UTC.

a number of conversions and tidying processes are necessary to ensure a seamless read-in of the data to the flight planning engine. Fig. 7.5 illustrates this process, starting from the point of filtering for data valid at specific dates to the generated flight plans' comparison with the actual trajectory. Generally, this process is di-

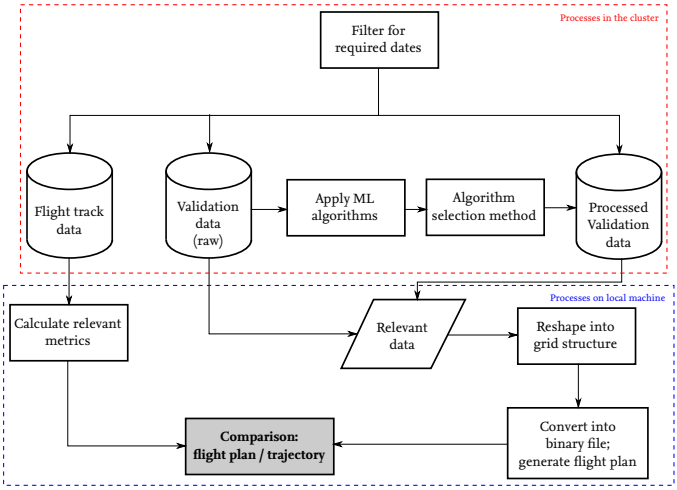


Figure 7.5.: Process for forecast weather data retrieval from the data cluster and further necessary processing steps.

vided into a part processed on the data cluster and another performed on a local machine. With the flight planning engine being a program running on a local machine, all necessary data will therefore eventually need to be downloaded to which. Since the data cluster holds significantly more processing power as compared to a single machine, as many as possible steps are performed in the former.
The process is initiated with a list of dates for which both a flight trajectory is to be retrieved, as well as a corresponding flight plan to be planned. These dates serve as a filter criterion for three separate databases; the FlightAware database, which stores flight tracks and the databases containing validation data. A distinction is drawn for the latter two between one database containing raw data (including forecast temperature data) and another containing the processed data from the trained algorithms and the algorithm selection process. The raw data covers all pressure levels from 850 to 150 Millibars (mbar), while the algorithms are only applied on the levels of cruise flights, between 300 and 150 mbar. Therefore, a data query is needed to be executed in both databases. The filtering query is realized in such a way that filtering for any weather data is only performed if any flight track data is actually returned from the FlightAware database.
With completion of the above-detailed processes, two separate data sets are outputted. For the flight track data set, an evaluation in regards to flight distance and time is performed. This aggregated information, together with a temporal and date key, is downloaded from the data cluster. All subsequent steps including the calculation of these two metrics from this point onward are therefore performed on a local machine. Downloaded weather forecast data is coerced into a gridded structure, in order to support a conversion of the data valid on a 0.5° lateral resolution to one of 1.25°, which is utilized by the flight planning engine. Conversion is realized using a 2-dimensional (2D) interpolation, which linearly interpolates between neighboring points. In the case of coordinates available in both grids (at e.g. N0° E0°), no interpolation is performed and the current data value retained.
Following this process, a final conversion of the data into a binary file format is performed. This binary file then serves as input to the flight planning engine, in which the flight plan is generated. Necessary input information are:

- Aircraft type
- Departure and arrival airport, in International Civil Aviation Organization (ICAO) format
- Date and time of departure
- Weather data binary file (optional)

- Activation of Eurocontrol RAD routes (optional)

The first three pieces of information are minimally necessary to successfully generate a flight plan. Weather forecast data, as well as the setting to evaluate potentially restricted RAD routes, are optional settings. Omitting these nonetheless results in a flight plan being generated. In such a case, no winds are evaluated and the temperature on all pressure levels is assumed to be those defined in the International Standard Atmosphere (ISA). Geopotential data is not required as the altitudes of pressure levels are assumed fixed at nominal ISA levels and corresponding altitudes.
From the resulting flight plan, the relevant values concerning both metrics (flight distance and time) are determined and compared against those from the corresponding actual flight tracks. This process is repeated in the identical way for all four sets of flight plans, with the filter query modified accordingly at the start of the process illustrated in fig. 7.5.

7.6. Results

This section presents the results of the flight plans generated. Specifically, all generated flight plans' (see fig. 7.2) results on projected flight duration and distance are in a first step compared to the actual values recorded for the flights in consideration, as illustrated in fig. 7.5. Doing so yields a measure of discrepancy between the respective plan and the trajectory. Such a discrepancy in this work is defined as the measure of predictability of a flight plan. A discrepancy of zero represents the most ideal state, of which a flight plan exactly predicts the actual trajectory. This scenario represents the maximum possible extend of predictability. As such, it represents the most beneficial condition to both airlines and the wider ATM system as a whole.
The determined discrepancies of the four sets of flight plans can thus be defined as in the following:

- ΔC: the flight plan generated on the basis of the original forecast data, i.e. the control.
- ΔE: the flight plan generated on the basis of the modified forecast data, produced by the machine learning process developed in this dissertation. This flight plan represents the output of the experiment in question.
- ΔE_{best}: the flight plan generated on the basis of the data produced by the machine learning process herein, albeit with the most accurate prediction selected.

- ΔC_{best}: the flight plan generated on the basis of re-analysis data. Since this data is assumed to be the most accurate, this flight plan serves as a hypothesized best-case control.

Generating these discrepancies does not directly yield insight into the validation of the concept herein. Since the research question is whether the product of the machine learning concept (the flight plan) has a higher predictability than the original flight plan, the *difference between the discrepancies* is of interest. As such, a flight plan generated on the basis of the machine learning concept carries a greater predictability under the condition that

$$\Delta C - \Delta E > 0. \tag{7.1}$$

A value < 0 in turn indicates that the experiment flight plan carries lesser predictability than the original one, whereas a result $= 0$ shows no difference. For all results in this chapter, equation 7.1 is utilized. The unit changes depending on the criteria being evaluated (see section 7.3 for details). Results are however only discussed for the criterion of flight duration, as no differences can be observed for flight distance.
The following sections presents and discuss the results from the three flights, as detailed in section 7.4. These first focus onto discussing results on a categorical level, i.e. whether equation 7.1 is greater, equal or less than 0. In a second section, these results are refined to display the actual differences. Throughout both sections, results for all flight plans in comparison to the control flight plan are discussed.

7.6.1. Hypotheses

The following hypotheses are discussed and verified/falsified in section 7.6.6:

1. **Hypothesis H7:** Flight plans generated on the basis of the herein modified forecast data will vary in their predictability depending on the world region the flight travels over.

2. **Hypothesis H8:** The results of best-case experiment flight plans will be of greater benefit in terms of predictability than those of the experiment flight plans.

3. **Hypothesis H9:** Flight plans generated on the basis of re-analysis data will feature the least discrepancy to the actual flight trajectory among all sets of flight plans generated.

7.6.2. Differences in flight duration discrepancy: categorical results

Results of the experiment flight plans are illustrated in fig. 7.6. These are divided into three categories, as described by the details given in section 7.6 and based on eq. 7.1. Shown are the respective results for each of the three haul types, or flights. Each category result is presented as a percentage of the haul type's total number of flights.

Short haul flights indicate a tendency for the experiment flight plans to generally have a greater planned-to-actual discrepancy than the corresponding control. Lesser discrepancies can only be observed for 15.7% of flights. This trend cannot be repeated for medium-haul flights, for which the vast majority (63.9%) of flights do not show a change of discrepancy between experiment and control. In this scenario, cases of lesser discrepancies grow to just under 29% of all flights. For long haul flights, the prior two trends can also not be repeated. In this case, the relative majority of flight instances exhibit less experiment discrepancy as compared to the control. A trend can be read from these results: the percentage of lesser exper-

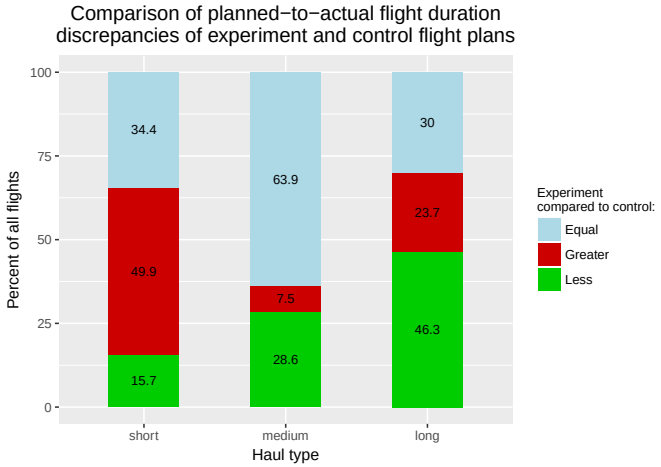


Figure 7.6.: Comparison of discrepancies in flight duration for experiment and control flight plans. Results are displayed per category; an experiment flight plan discrepancy lesser than of the original flight plan is defined as beneficial. Shown are the results for the three flight routes outlined in 7.4.

iment discrepancy flight plans grows with flight distance. Given the overall trend from the algorithmic testing and algorithm selection method (see chapter 6), a region of generally positive algorithmic performance is identified to be South East Asia. The short haul flight which passes through this region, does not mirror the benefits indicated by algorithmic prediction results. Instead the contrary is true, with close to 50% of experiment flight plans having a greater discrepancy than the control. Algorithmic results for the medium haul flight do not show the beneficial results as exhibited by locations in South East Asia. Based on this knowledge, it can naturally be assumed that this translates into results less beneficial than those of the short haul flight. However, this is not the case as the number of lesser experiment discrepancies almost doubles from short to medium haul. This trend continues for the long haul flight.

One possible explanation for this behavior may be that the longer the flight distance, the less impact an inaccurate algorithmic prediction may have. Assume that a forecast data set generated by the concept herein that is very far from the true value (i.e. very inaccurate) will lead to the flight planning engine generating a forecast (at least for the coordinate location concerned) that too it very inaccurate. If the distance between the departure and arrival cities is shorter, the number of possibilities to fly along a different route also decrease. If a number of forecast data points have poor accuracy, the flight planning engine will be more and more restricted in the number of possibilities to find a more economical flight route. While it obviously does not have the information on which location's forecast is more accurate than another, the chance of finding a route that is closer to the true trajectory is increased. This explanation is supported by the further percentage increase in lesser experiment discrepancy for the long haul flight. Fig. 7.7 illustrates a schematic representation of this explanation. A shorter flight (in terms of flight

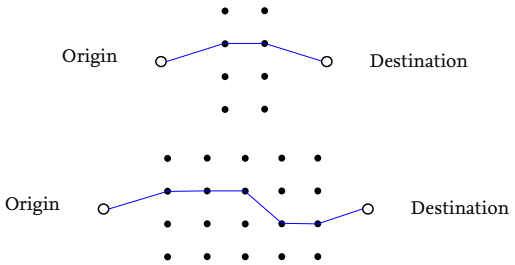


Figure 7.7.: Schematic comparison of the differences in the number of points evaluated by the flight planning engine, between shorter (top) and longer flights (bottom).

distance) results in a lesser number of coordinate locations upon which the flight planning engine determines the route with minimum cost. In turn, a longer flight yields more locations on which a solution is to be found. If a number of these locations bear a particularly inaccurate forecast value, the likelihood of this affecting the short flight will be greater than with a longer flight, as the number of possible solutions are less. The difference between experiment and control might therefore likely be either very low or very high. Thus, the spread of discrepancy comparison values will likely qualitatively differ between short and medium on one side long haul on the other. As such, more extreme negative and positive might occur with a decrease in flight distance. The spread of values might therefore resemble more of a normal distribution the greater the flight distance is. This assumption will be further examined and discussed in section 7.6.4.

Categorical results for the best-case experiment scenario

The best-case experiment, as illustrated in fig. 7.2, is the flight plan generated on the most accurate prediction. In this case, the algorithm selection process is coerced to select the prediction generated by the eight machine learning algorithm that is closest to the true value of the corresponding re-analysis data. This is a process based on posterior knowledge, i.e. knowing what actually is the most accurate prediction. Thus, it is not a realizable working order for the concept herein. However, the assumption can be tested that the more accurate the input data, the more beneficial the impact onto flight planning is. Hence, the assumption is that the result of this comparison will be more beneficial than the prior comparison based on the experiment and control flight plans (see fig. 7.8). Results for the short haul flight verify the assumption that if the algorithm selection method were to always select the most accurate prediction, the flight planning predictability were to increase. This can be read from the increase by the slight increase in percentage for the lesser discrepancy. In addition, the percentage of flight plans with greater discrepancy than the control shrinks from 49.9% to 8.3%.
However, this trend is not confirmed by medium and long haul flights. Both sets show a decrease in beneficial results i.e. less difference and a concurrent increase in the number of non-beneficial instances. Such a result indicates that the prior assumption does not hold true for longer-ranged or at least flights outside South East Asia. It is therefore of interest as to how these results in turn compare against flight plans that are created on the basis of re-analysis data, presumably the most accurate depiction of the atmosphere.

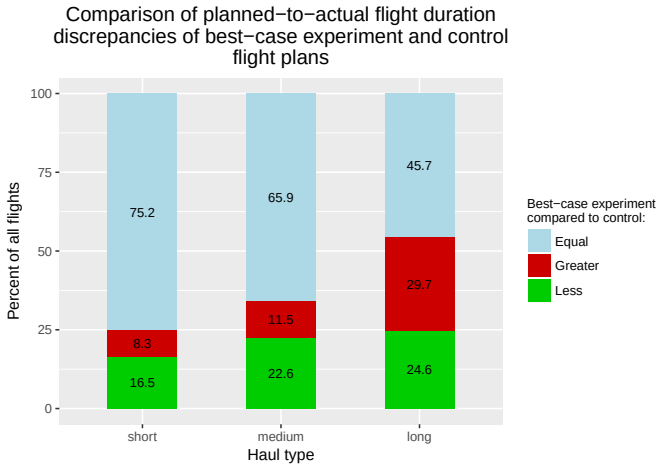


Figure 7.8.: Comparison of discrepancies in flight duration for best-case experiment and control flight plans. In this scenario, the best-performing algorithm is selected in every case. Results are displayed per category.

Categorical results for the best-case control scenario

As indicated in prior section 7.6.2, it is not only interesting to evaluate how a best-case scenario performs against the control. Rather, it is of further interest of comparing the best-case scenario against the supposedly best-case control. These, as outlined in section 7.2, are based on re-analysis and thus, the assumed most accurate data. Hence, the assumption is that flight plans generated on the basis of this data yield those with the lowest discrepancy between themselves and the actual flight trajectory. Fig. 7.9 illustrates the categorical results determined for this set of flight plans. Notable differences exist between these results and those from the best-case experiment. While for the short haul flight lesser differences decrease slightly by 0.2, all other results change by at least 0.9 percentage points. Of particular interest are the results for a greater difference, as these represent the undesired case. These feature values consistently greater than for the best-case experiment scenario. Results for the lesser difference between plan and trajectory also increase. Such an observation points to a greater scattering of the data. Either the data is very close to the actual true atmospheric values or it misses it on a greater magnitude than the best-case experiment.
Both experiment and best-case experiment data is the output of a prediction that

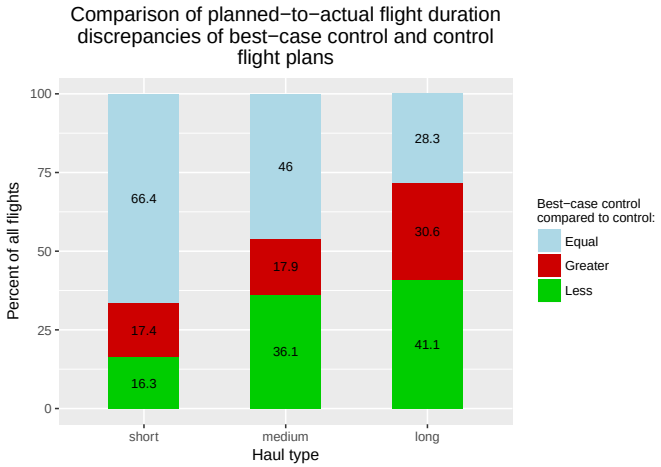


Figure 7.9.: Comparison of discrepancies in flight duration for best-case control and control flight plans. Re-analysis data is used to generate the best-case control flight plans. Results are displayed per category.

is calibrated with the target function being the re-analysis data. This means that algorithmic predictions will have, ideally, less discrepancy between itself and the re-analysis than the original forecasts'. The re-analysis data, while assumed to be the most accurate description of the atmosphere, is also only a computed value with an unknown discrepancy to the absolute true value. Therefore, it is likely that any predictions generated herein are nearer to the true values than the re-analysis. This assumption agrees with the recognized trends in fig. 7.9, especially for the category of greater difference.

7.6.3. Changes in results between cases

On top of the categorical results discussed in the prior section, it is of interest how each category is comprised of. As one flight plan is generated for each case (for every date tag the flight was actually conducted), a flow for each instance can be defined. This flow effectively describes the changes in category between the three cases. Of particular interest are the changes between experiment and best-case experiment and between the latter and the best-case control. A change between the first two cases is interesting in this sense as the best-case experiment is assumed to yield more beneficial results. Changes between best-case experiment and best-

case control are insofar of interest since the latter is assumed to represent the most accurate description of the atmosphere and is thus assumed to yield the most beneficial results.
An alluvial diagram expressing this relation for short haul flight GIA867 is illustrated in fig. 7.10. Depicted are the flight's results for the three cases, as discussed

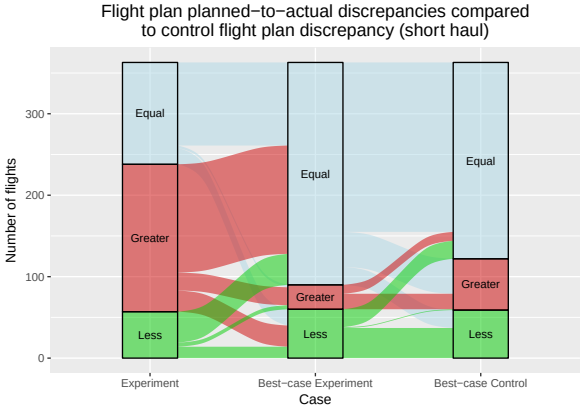


Figure 7.10.: Alluvial diagram showing flows between the three cases' categories, for flight GIA867.

in section 7.6.2. The composition of each category from the prior case is illustrated. Experiment and best-case experiment cases are different in that the latter is modified to represent the hypothetical case of the most accurate prediction being selected. In the case of the short haul flight, it was shown in section 7.6.2 that results for the best-case experiment showed a doubling of the number of instances with equal discrepancies. Fig. 7.10 adds the additional information which experiment's categories contributed in what magnitude. These shows that the growth in the number of equal discrepancy instances (between experiment and best-case experiment) stems almost entirely from instances tagged with a greater discrepancy in the experiment case. This observation can be interpreted as follows: since the most accurate predictions are selected in the best-case experiment, this increase in accuracy is translated into a decrease in the number of greater discrepancy instances.
Changes between best-case experiment and best-case control show that a majority of instances with a greater discrepancy (in the best-case control) exhibited equal discrepancy in the best-case experiment. This observation, together with the recorded

doubling of instances with greater discrepancy show that the overall flight plans carry less predictability if planned with re-analysis data. While no proof can be provided at this point, this may be an indicator that the re-analysis data's error is greater than that of the hypothetical best-case experiment's data.

Results for medium haul flight TUI2148 paint a picture contrary to that found in the prior flight. Running flight plans in the second case does not improve predictability. While the number of lesser discrepancies decrease, the number of greater instances increases. The conclusion that can be drawn from this behavior is that the presumed most accurate prediction may be the most accurate in terms of re-analysis data. Since this data set by nature carries a magnitude of inaccuracy, the resulting prediction may be farther from the true value than the re-analysis data. Results generated on the basis of re-analysis data show ambivalent results. While the number of instances with lesser discrepancy increase, the number with greater discrepancy increase. Both categories' increases almost entirely stem from the equal category. An explanation for this behavior may be that the added accuracy of re-analysis data leads to a beneficial effect, while also surpassing the true weather values. The latter behavior may lead to the increase in greater discrepancy instances.

The third group of results, for DLH778, shows yet another different set of trends than the prior two flights. Similar to TUI2148, the trend between experiment and best-case experiment results can be observed. Different is the fact that the results of the best-case control are worse than of the experiment, with an increased number of greater discrepancy and a decrease in lesser discrepancy instances. Once more, the reason for these results may be the inaccuracy of the re-analysis data to the true weather condition. Since the Machine Learning algorithms utilize this data as the target function, any inaccuracies are carried over into the algorithms' learned behavior.

7.6.4. Histograms of differences in flight duration

This section presents and discusses the difference in discrepancy values for all three flights. Shown are those of the comparison between experiment and control. Each flight's results in the following three figures (7.13, 7.14 and 7.15) bears a number of recognizable patterns, which are to be identified and interpreted. The short haul flight, as discussed in the section on categorical results, bears a tendency towards negative results. Most common negative results are recorded for a discrepancy of 60 seconds and another minor peak at 130 seconds. The most negative differences are recorded between 160 and 180 seconds. Further observations are the small-scale occurrences between 0 and -60 seconds of difference, which

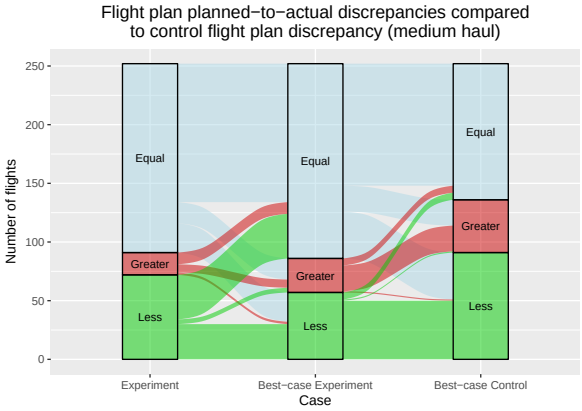


Figure 7.11.: Alluvial diagram showing flows between the three cases' categories, for flight TUI2148.

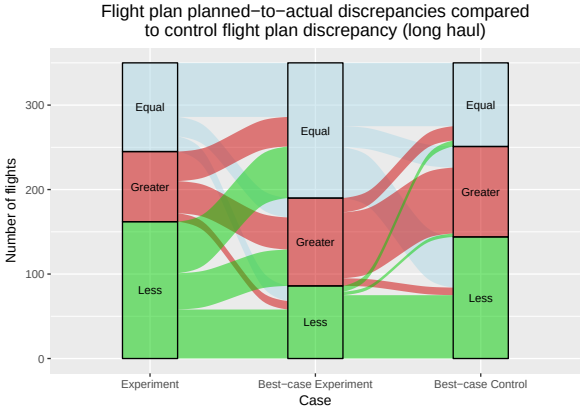


Figure 7.12.: Alluvial diagram showing flows between the three cases' categories, for flight DLH778.

in a similar manner appear on the positive side between 0 and 60 seconds. Two minor peaks can also be identified on the positive side of differences, at 60 and 125 seconds.

The fact that the extremes of differences all lie between +2 and -3 minutes and a skewness towards the negative side of differences indicate that the effect of modi-

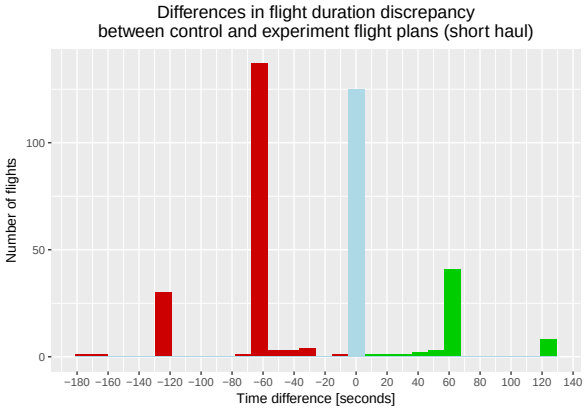


Figure 7.13.: Histogram of duration differences between control and experiment flight plans to the actual trajectory, for the short haul flight GIA867.

fied weather data on the flight plan does not generate a favorable benefit over the control flight plan. In turn, this result can be interpreted that in this world region, it is advisable to either continue to utilize the original forecasts or ensure an increased algorithmic prediction performance of the algorithm selection method. Medium haul flight TUI2148's results in fig. 7.14 show a different picture, as the overwhelming majority of cases remain at 0 difference. The most extreme negative occurrence is located at -240 seconds. As the number of positive differences outnumber negatives, the experiment can be expressed as beneficial for this flight, with gains of up to a maximum of 120 seconds. Long haul results in fig. 7.15 illustrate the set of results with the smoothest edges. While the most extreme negative instances are recorded at -600 seconds, the overwhelming majority of negative instances occur at or before the -180 second mark. In contrast stand the positive results, of which the majority of instances are recorded at or before the 180 second mark.

These results from the histograms of the three flights show that a magnitude of change, whether positive or negative, is within the single-digit boundary. In addition, the peak of instances with no change (a difference of zero between control and experiment) shows that the experiment's impact is limited. While this on one hand decreases the beneficial effects, it also shows the method's reliability to not produce extreme outliers. Definition of this is of particular importance, as the flight plan directly leads to the calculation of fuel. Calculation of a completely

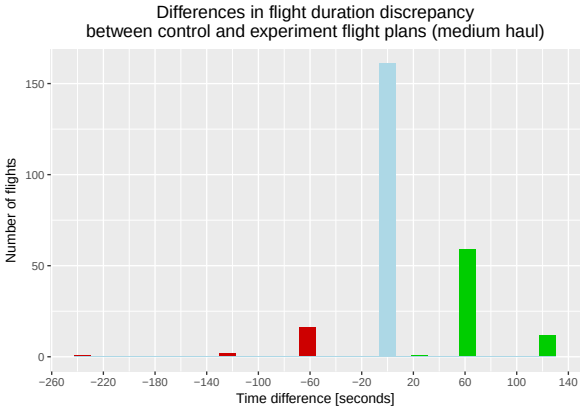


Figure 7.14.: Histogram of duration differences between control and experiment flight plans to the actual trajectory, for the medium haul flight TUI2148.

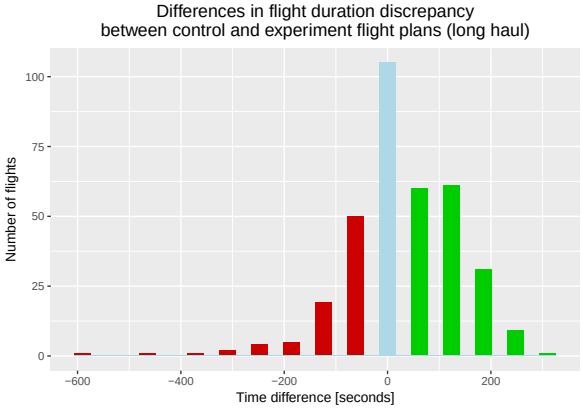


Figure 7.15.: Histogram of duration differences between control and experiment flight plans to the actual trajectory, for the long haul flight DLH778.

unrealistic amount of fuel would not only decrease efficiency, but infringe on the flight's safety. Therefore, while the histogram results are ambivalent, the scattering of which may be in an acceptable range. This reasoning is derived from the allocation of contingency fuel per ICAO Doc 9976 guidelines [5]. In this context, a contingency of 5% of trip fuel is allocated to unforeseen factors. Assuming an

equal spread of fuel across the flight, the most negative extremes for the respective flights can be used to determine the percentage of the total flight time. 3 minutes of GIA867's flight time of 3 hours and 10 minutes equal 1.6% of the total flight time. For TUI2148 this number is equal to 1.5% and 1.4% for DLH778. While these values all are less than the 5% contingency, more research is needed to be done to examine whether these results produced herein are justifiably covered by this fuel allocation.

7.6.5. Patterns in discrepancy occurrence

Prior flight plan results have shown cases in which benefits of the experiment of concern can be identified. Equally interesting is the question of whether one is able to distinguish a flight plan with less, greater or equal discrepancy than the control, at the time of flight planning. If a user is able to determine the likely category at the time of planning, a flight plan with greater discrepancy (i.e. less predictability) could be identified and rejected. For this to be realized, the results herein have to show a pattern by which flight plans can be divided by category. In this section, two possible options are examined. The first involves arranging results by month and examining a possible seasonal relation. In a second step, the output flight plans' planned fuel and flight duration values are plotted in order to investigate whether patterns can be identified.
For this section's discussion, only results for the experiment and best-case experiment cases utilized. Since in an operational scenario, the re-analysis data for a future flight is not available at that time, a flight plan generated with this data is not operationally realizable.

Results by month

When split by months, experiment results illustrated in fig. 7.16 for the short haul flight show that with the exception of January all months feature instances pertaining to all three categories. Compared against results of the best-case experiment case, a significant drop in the number of greater discrepancies can be recorded. In this case, the month of October on top of January is seen as featuring no instances of greater discrepancies. A pattern of seasonal performance such as that observed in results on algorithm performance (see fig. A.23), cannot be identified for either set of results. While some months exhibit a greater number of flights with greater discrepancy, the spread nevertheless covers 10 months.
Similar observations are recorded for the other two flights of concern, as illustrated by figures A.25 and A.26. Beneficial experiment instances consistently outnumber those of greater discrepancy. However, a seasonal dependency on flight plan predictability cannot be drawn from these.

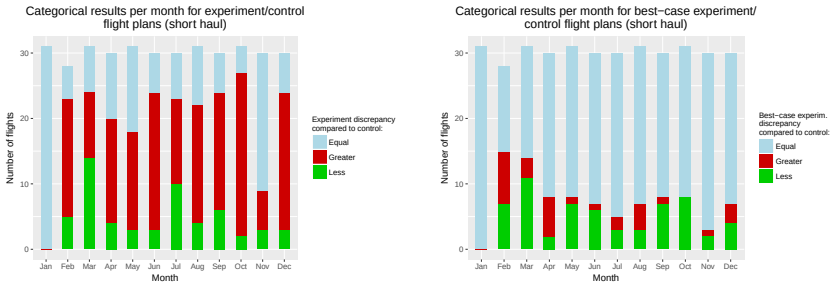


Figure 7.16.: Monthly results for the experiment (left) and best-case experiment (right), for short haul flight GIA867.

Results by planned fuel and flight duration

Planned fuel and flight duration are only two of a multitude of different values outputted by a flight planning engine. In this research, a determination is able to be done whether a flight plan carries greater, equal or less discrepancy to the actual trajectory than a control one. Thus, the information is known which of all flight plans generated are beneficial or not to the flight planning process. By coupling this information with the information that a flight planning engine generates, i.e. required fuel and flight time, patterns might be retrieved. Doing so asks the question whether one is a priori able to specify whether the generated flight plan is more or less beneficial than the control. To investigate this question, fig. 7.17 has been generated for the short haul flight. Illustrated are the results of flight plans regarding their planned duration and fuel. A third dimension is provided, as to whether each carries greater, less or equal discrepancy than the control flight plan. For every category, a local regression fit is provided on top to capture the underlying patterns of these.

The fact that all three regressions are located in the same vicinity without clear discrepancies between them show that an underlying pattern in these two dimensions does not exist. A consequence of this is that a measure of confidence in the added predictability of a flight plan cannot be determined prior to the flight's conduct. This conclusion is also valid for results of the best-case experiment, as illustrated in fig. A.27 and the over two flights (figs. A.28 and A.29).

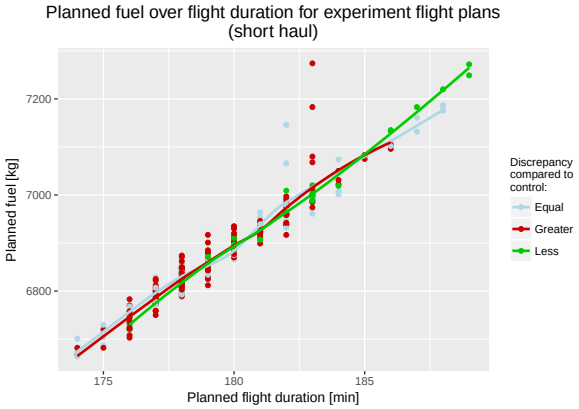


Figure 7.17.: Experiment: Planned fuel and flight time for short haul flight GIA867.

7.6.6. Hypotheses

This section revisits the hypotheses stated in 7.6.1 and determines whether the above findings result in a verification or falsification of these.

Hypothesis H7: *Flight plans generated on the basis of the herein modified forecast data will vary in their predictability depending on the world region the flight travels over.*
Results shown in fig. 7.6 show varying degrees of discrepancies for all three categories defined. Especially the difference observed for the short and medium haul flights shows that the flight plan's predictability depends heavily on the world region. This observation further agrees with the location-dependent prediction performance of the algorithms trained in this work.
Thus, hypothesis H7 cannot be rejected.

Hypothesis H8: *The results of best-case experiment flight plans will be of greater benefit in terms of predictability than those of the experiment flight plans.*
Results for the short haul flight are in agreement of this statement, as illustrated in fig. 7.10. However, results for the two other flights actually showcase a growth in the greater discrepancy category, indicating less accurate flight plans and and such ones of lesser predictability and hence, benefit. The reason for this behavior may lie in the fact that algorithmic prediction performance is greatest in general in tropical latitudes. The thus added accuracy might translate into a more

accurate flight plan. Forecast accuracy improvement in general is less outside this area, which may point towards a similar prediction performance among algorithms trained. As such, the algorithm selection method may still successfully select the algorithm of greatest prediction performance. This prediction's accuracy however may not much differ from those of the other algorithms. This hypothesis can solely be verified for one of the three flights and cannot be utilized as a generally true statement. Thus, hypothesis H8 is falsified.

Hypothesis H9: *Flight plans generated on the basis of re-analysis data will feature the least discrepancy to the actual flight trajectory among all sets of flight plans generated.*
The same results as those discussed in hypothesis H8 are used to determine a verdict on this hypothesis. Since re-analysis data is assumed to be the most accurate description of the atmosphere's meteorological state, the assumption is further extended that this most accurate data will also translate into most accurate flight plans. Results on this hypothesis are ambivalent. The medium haul flight shows the greatest percentage of beneficial instances recorded with the best-case control flight plans. However in that case the number of instances with greater discrepancy is the greatest when compared between the three cases. In the case of the long haul flight, the highest percentage of beneficial instances is not found with the re-analysis flight plans, but rather with those of the experiment. Lastly, beneficial instances of the short haul flight are almost constant throughout all cases. However, flight plans generated on re-analysis data do not account for the least number of non-beneficial flight plans. Due to these observations, hypothesis H9 is falsified.

7.7. Summary

This chapter presents the approach undertaken for the validation of the system for forecast uncertainty prediction. For this, the methodology is outlined, including the definition of experiment and baseline scenarios. Performance criteria by which the validation is performed, is presented. Out of these, two criteria are identified to be suited for the validation purpose. Three flights are further selected for validation, in order to have one available each for short-, medium- and long-haul. The data handling process for data retrieval and coersion to a binary format for the flight planning engine is outlined and described. Lastly, the results generated from the flight plans generated are presented and discussed.

8 Conclusion and outlook

This chapter concludes this dissertation by providing a summary of the work and results. An outlook is furthermore provided, outlining a possible number of next steps to improve and evolve the concept proposed herein. Lastly, a hypothetical embedding of this concept in an operational environment is presented.

8.1. Conclusion

This conclusion is divided into three parts. First, a summary is provided for the core idea, research gap and the resulting proposed concept. A realization is then presented, after which the results of the different evaluations are provided.

8.1.1. Conceptualization

Core to the idea of this work is the question of whether Big Data analysis technologies are able to improve the effectiveness of the flight planning process. This mainly involves the capability of handling large amounts of data, including doing so with a high degree of parallelization. Further, given the possible parallelization, machine learning is utilized in this context for prediction generation based on the algorithms trained on this data.

The above requirement involving the focus on flight planning necessitates research towards the ingestion of data in this process. While a large and varying number of data sources are involved in planning a route, weather is identified to yield the greatest inaccuracies in trajectory prediction [6]. In addition, weather processes by nature are highly complex and difficult to correctly forecast. A prime method for weather forecasting are numerical simulations running on models imitating the atmospherical physics. However, as the term *models* implies, these only describe an approximation of the true state. Approximations thereby yield uncertainties in forecasts, in effect inaccuracies between the forecasted and the actual true value. Translated to flight planning, these uncertainties in turn yield inaccuracies in the predicted values of the flight plan.
As such, the identified goals of this research are to evaluate whether:

1. Weather forecast uncertainties can be predicted using a data-centric machine learning approach by training algorithms to predict these;

2. The generated predictions yield a benefit to the flight planning process.

The first requirement specifically places emphasis on the use of machine learning and not a deterministic numerical simulation. Significantly less focus has been drawn to the method of forecasting by statistical than by deterministic means. Advances in technology have in addition opened the door to the efficient processing of large-scale data sets. While the trained algorithms will also be an approximation, it is of interest whether these produce a result that is closer to the true weather state than conventional forecasts. Finally, the usefulness or benefit of utilizing this method over the current state of the art is evaluated.

8.1.2. Realization

The flight planning process searches for the most cost-optimal route among a multitude of different possibilities arising from the wealth of waypoints and airways. As such, a broad lateral coverage of weather data points is required. In addition, vertical levels, temporal forecast steps and different data variables create an environment with a high number of partitions. Since statistical relevance is fundamental in a data-centric approach, the amount of weather data valid at each of these partitions is required to be as large as possible. To meet this requirement, a time frame covering 9 years and 8 months from 2006-11 through 2016-06 has been curated for this purpose. Of this data, the original forecasts, as well as the correspondingly valid re-analysis data is utilized. While the former represents the data currently ingested by flight planning engines, the latter is considered the most accurate description of the atmosphere across such a broad coverage. The resulting data set's size hence requires the utilization of a Big Data cluster, on which the data is cleaned, handled and processed. Boeing Research and Technology – Europe's data cluster, running on Apache's *Spark* framework is utilized in the course of this work. Especially the latter's possibility of running multiple processing threads in parallel allows the processing of a large number of partitions in a feasible time frame.

A number of machine learning algorithms are selected, each with a different advantage and strength. As the nature of weather is highly complex, one single algorithm is assumed to be incapable of capturing every aspect. Rather, a system is realized in which eight different algorithms are each trained on the same partition's data set. These are then stored in the Big Data cluster. Using a test data set, each algorithm's prediction performance is determined. A second algorithmic layer is then realized, which utilizes these test results. Given an arbitrary input forecast, this layer utilizes these results to determine the closest data instance. Doing so yields the information of which algorithm generated the most accurate prediction, hence selecting

the most likely best algorithm for this given input forecast. For this *algorithm selection method*, another data set covering a year of data is curated, upon which this method's prediction performance is quantified. The output of this second evaluation then serves as the input to the flight plans generated in the concept's final validation.

8.1.3. Evaluation and flight plan validation

Due to the realized structure described in section 8.1.2, the evaluation is needed to be divided into three separate parts. First, using a split of the available data, the algorithms are trained and subsequently tested in their prediction performance. These results show that a high locational correlation exists for every algorithm. Results specifically show, irrelevant of the algorithm examined, that the best prediction performance is generally found in areas approx. south of 17° northern latitude. Coherent geographical patterns of the respectively best-performing algorithm can be identified, with these scoring up to a maximum of 44% more accurate predictions over the original forecasts. In the majority of cases, seven out of eight algorithms are identified to yield favorable results.
The algorithm selection method is realized as a k-Nearest-Neighbor algorithm. Based on an arbitrary forecast value, the distance of such to every stored test result forecast is calculated. In this way, the likely most accurately-predicting algorithm is determined and utilized for this current forecast. This algorithmic layer produces ambivalent results, with lateral patterns of beneficial prediction performance in similar vicinities as retrieved with the test results. Algorithmic prediction performance however varies significantly depending on the forecast lead time.

For the validation of the concept in the context of flight planning, a total of four different types of flight plans are created per flight. Three flights are focused on in this work. Each of the four flight plans are compared to the actual trajectory, in order to calculate the discrepancy, which serves as a metric of predictability in this research. These discrepancies are compared against each other, thus yielding a determination of whether the experiment (a flight plan generated with the algorithms' predictions) is more beneficial than the control (a flight plan generated on the basis of the original forecasts). The third flight plan is produced with the hypothetical setting that the most accurate algorithm is consistently being selected, in effect representing the upper boundary of algorithmic prediction performance. Re-analysis data serves as the weather input data for the fourth flight plan. These two sets respectively represent the hypothetical best the concept is able to achieve and the anticipated most accurate data available.
Results indicate a heavy dependence on the area of flight operations. The great-

est benefit of using the algorithm selection method's hypothetical best-case is observed for tropical regions. Results from the other two flights show a greater percentage of instances exhibiting a benefit of the method to flight planning predictability. Consistent throughout all three flights is the observation that based on the current results and insights, no reliable prior prediction on whether a flight plan is more or less beneficial than the status quo can be produced.

8.2. Outlook

This work's outlook relies on three conclusions drawn from the three sets of results discussed herein. Results for the trained algorithms show improvement in accuracy over the original forecasts. This is true even for low-complexity algorithms, such as the linear regression. In order to further move towards an eventual operational realization, a greater number of algorithms need to be trained and evaluated. On top of the number of algorithms, the number of locations should be maximized to cover the entire globe. For this process to run efficiently, the current system's runtime behavior needs to be improved significantly. Although algorithmic training and testing is essentially a batch process running only once in a time frame of months, this process shall ideally not require more time than between publishing of forecasts every six hours. Reason for doing so is the hypothetical scenario that a complete re-run of all algorithms is needed prior to receiving the next forecast.

Another focus of this work is the realization of a second algorithmic layer aiming to select the best suited algorithm for an arbitrary forecast, based on similarity expressed through a distance function. This process needs to be improved in its accuracy or removed if no improvements can be realized. A strategy for potential improvement of the process would be to find each partition's k number of neighbors that yields the best results. Another option is the evaluation of whether a different algorithm would generate a better level of performance. Irrelevant of the strategy, a reworked second algorithmic layer must not lead to prediction performance dropping, as compared to only utilizing a single trained algorithm.

The flight planning process should also be examined in greater detail. Results discussed herein concerning the impact of the machine learning system on flight plan prediction performance have consistently shown a seemingly normal distribution of results. Such an observation calls into question the scale of impact that weather forecasts hold in flight planning systems. Prediction performance improvement exhibited by the algorithms could be observed of up to 44% over the original forecasts, especially in areas of South East Asia. Flight plan results could not replicate the benefits recorded in the predictions. Based on this finding and the fact that the weather grid utilized in the flight planning system is coarsely-

grained, the hypothesis is rejected that weather data plays a major role in influencing the flight planning process.

A Appendix

A.1. Algorithmic test results

A.1.1. Linear regression

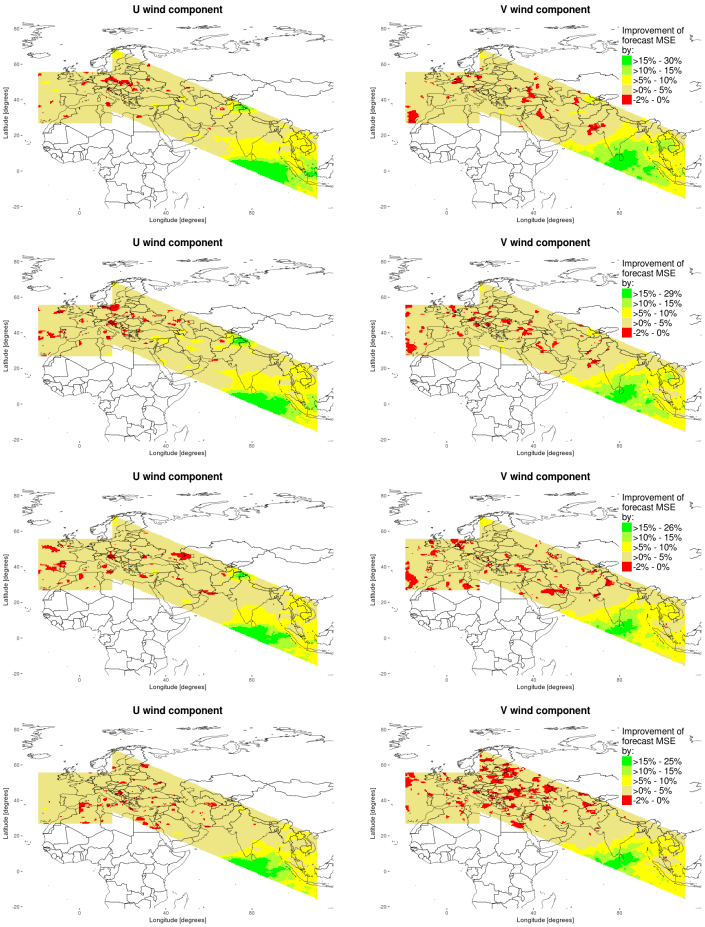


Figure A.1.: MSE test results for the linear regression, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.1.2. 5th degree polynomial regression

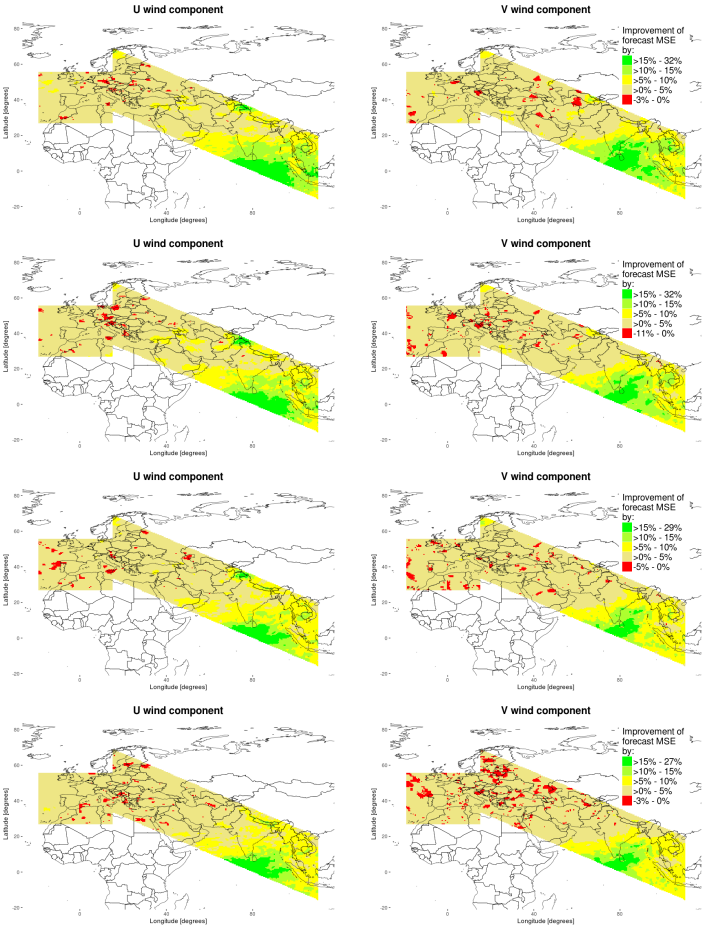


Figure A.2.: MSE test results for the 5th degree polynomial regression, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.1.3. 10th degree polynomial regression

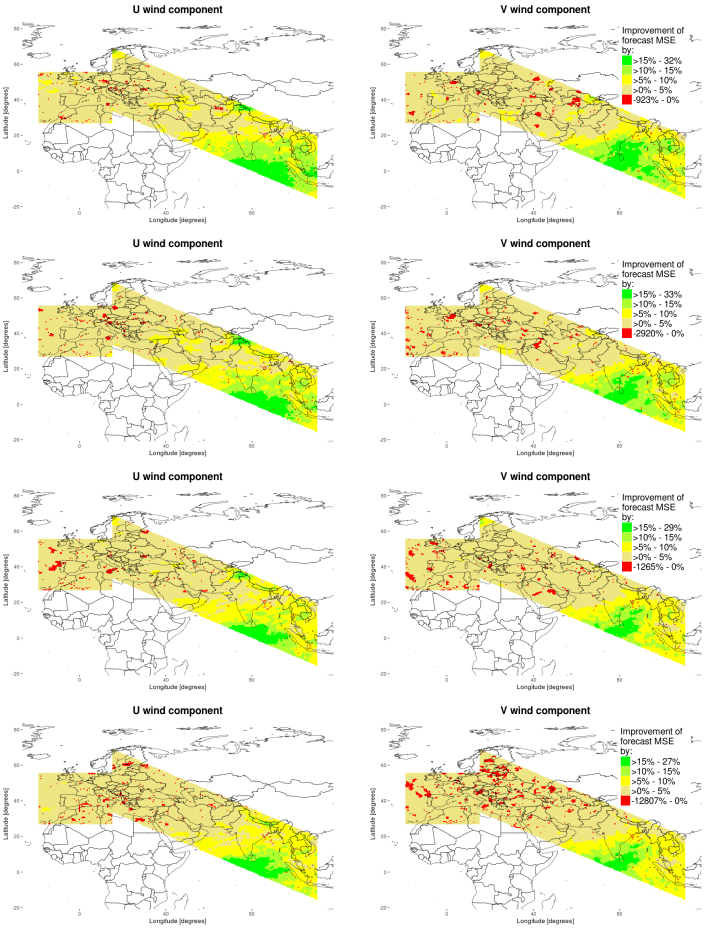


Figure A.3.: MSE test results for the 10th degree polynomial regression, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.1.4. 15th degree polynomial regression

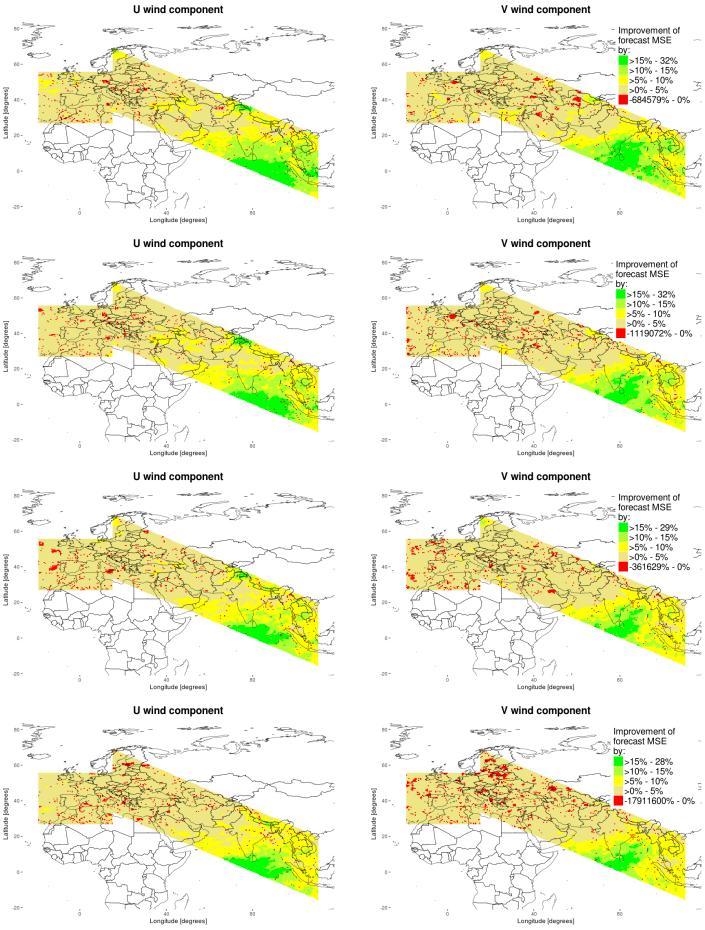


Figure A.4.: MSE test results for the 15th degree polynomial regression, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.1.5. Support Vector Machine (SVM)

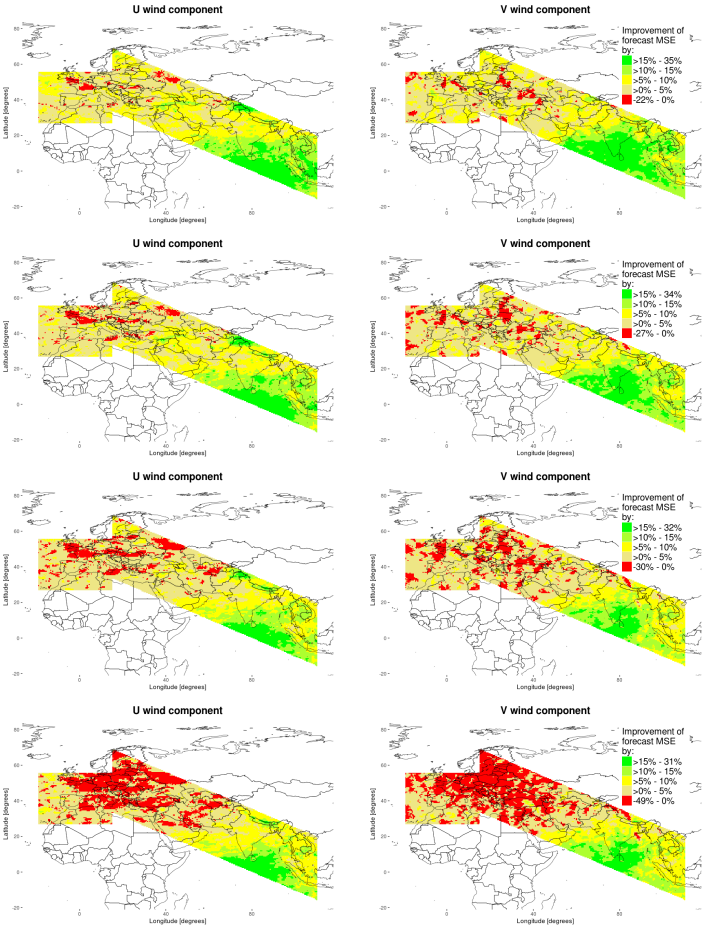


Figure A.5.: MSE test results for the SVM, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.1.6. Decision Tree

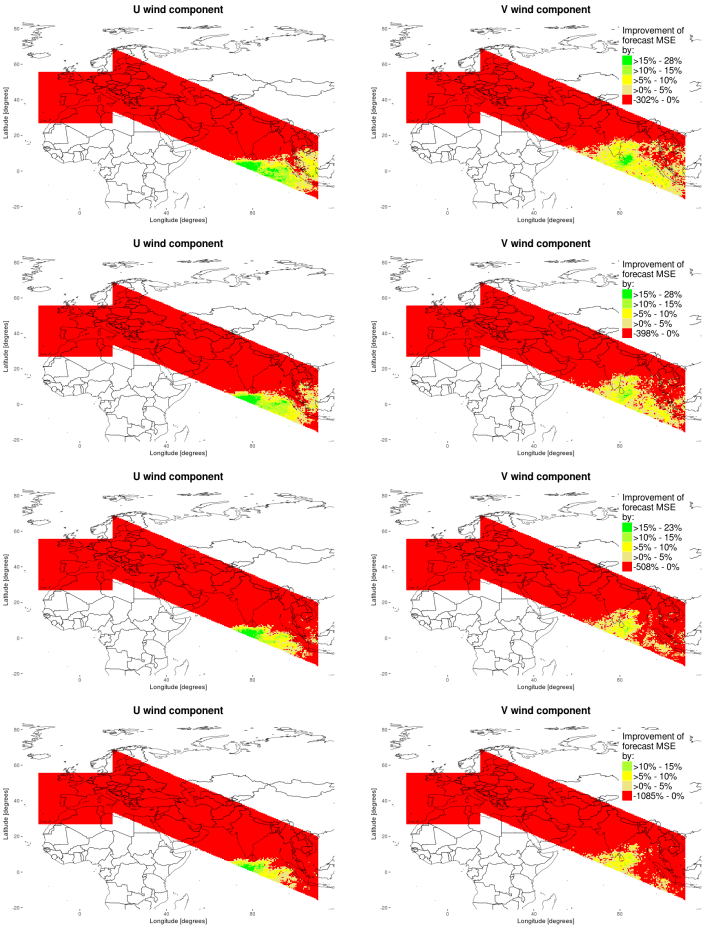


Figure A.6.: MSE test results for the Decision Tree, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.1.7. Boosting

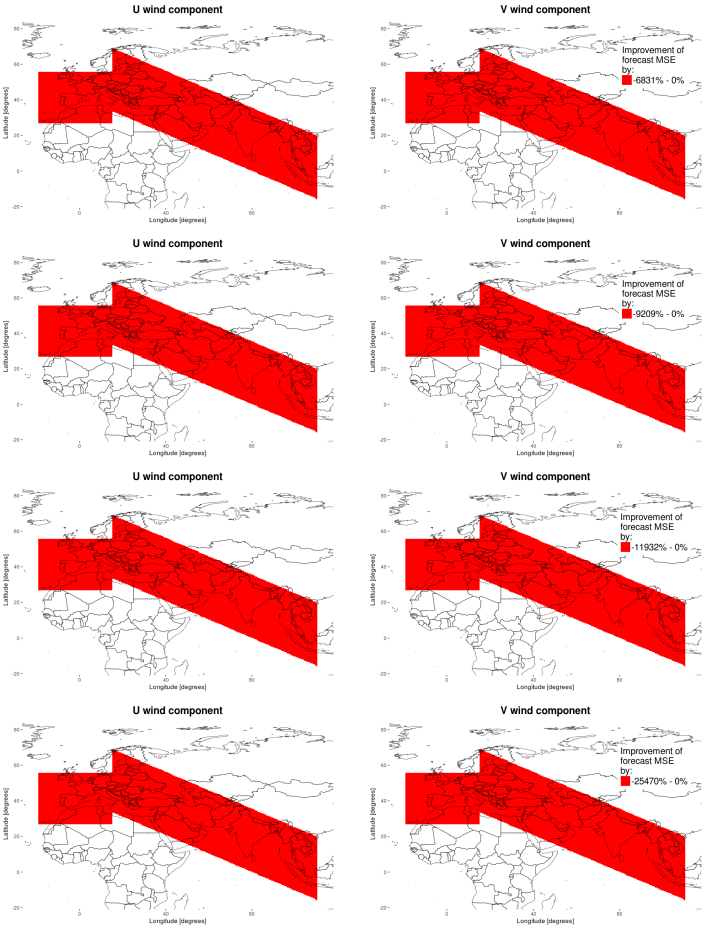


Figure A.7.: MSE test results for Boosting, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.1.8. k-Nearest-Neighbors (kNN)

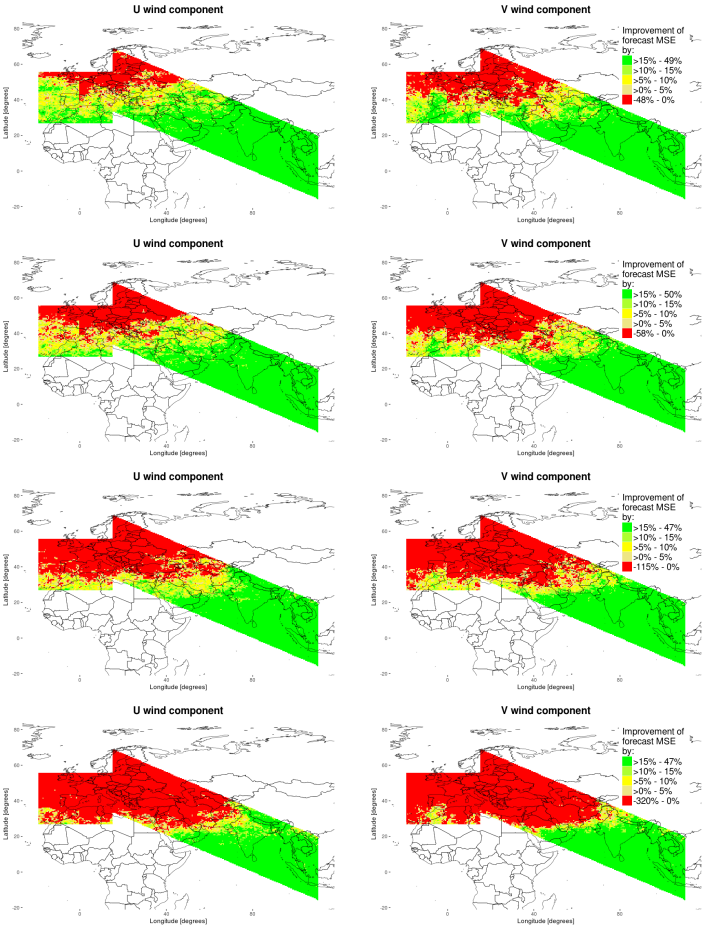


Figure A.8.: MSE test results for a kNN algorithm, applied on data at 200 mbar altitude, for U and V wind components for 18 h (top row) to 00 h (bottom row) forecast steps.

A.2. Time steps

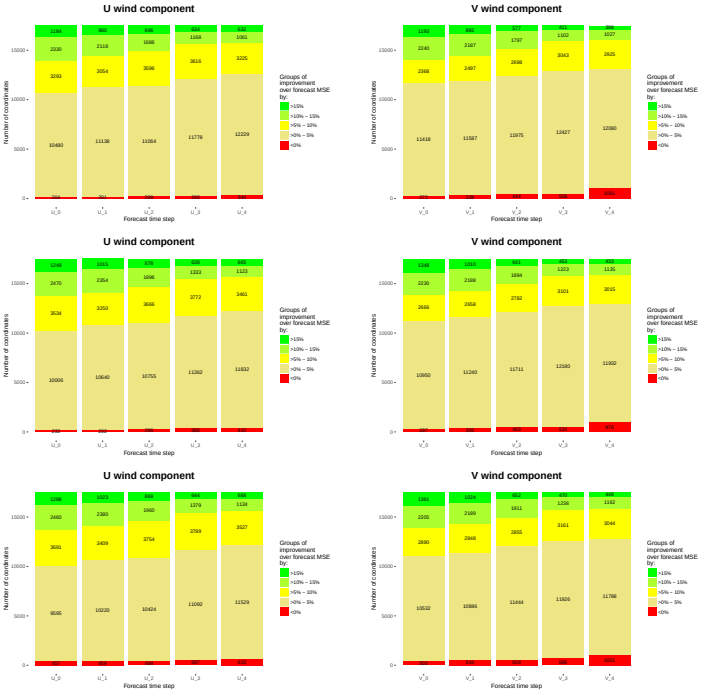


Figure A.9.: Number of coordinates in each of the five improvement categories for a testing of 5th, 10th and 15th-order regression for both wind components throughout all time steps for all higher-order polynomial regressions, at 200 mbar altitude. Forecast lead times range from 24 h (leftmost column) to 00 h (rightmost column).

A.3. Number of improving algorithms for other time steps

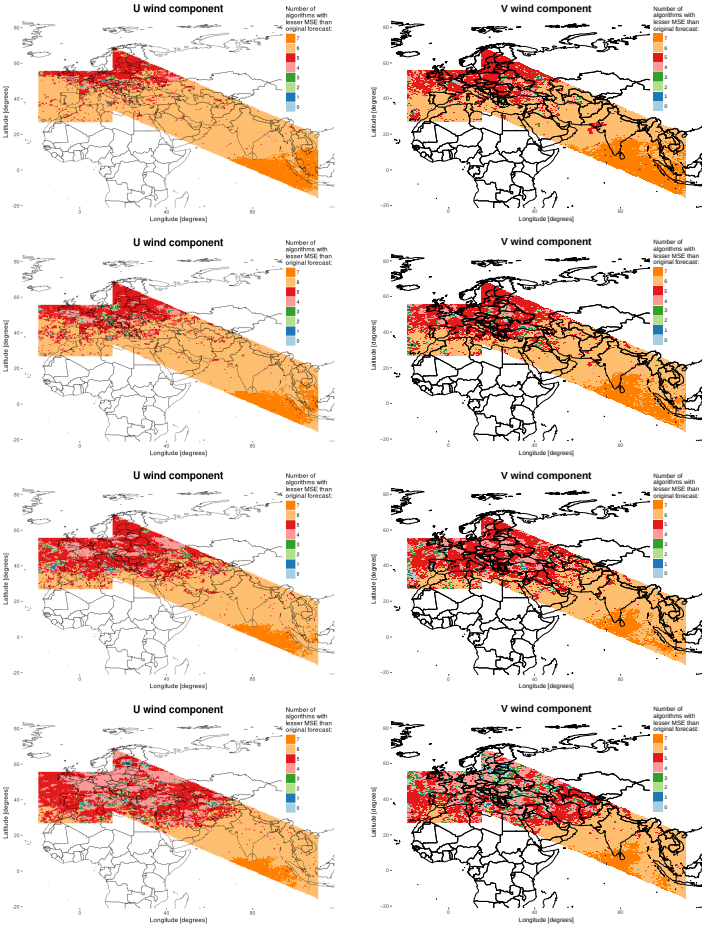


Figure A.10.: Number of algorithms with a lesser MSE than of the original forecasts for time steps 18 h (top row) through 00 h (bottom row). Illustrated are U and V wind components for 18 h through 00 h forecast steps, at 200 mbar altitude.

A.4. Best and worst algorithm for other time steps

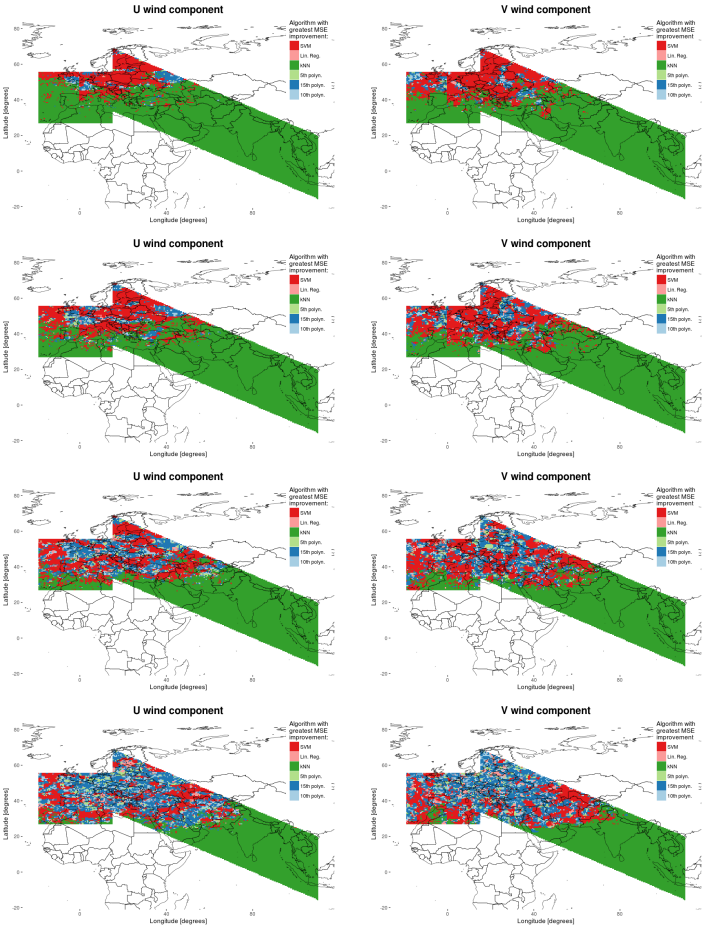


Figure A.11.: The best performing algorithms. Illustrated are U and V wind components for 18 h (top row) through 00 h (bottom row) forecast steps, at 200 mbar altitude.

A.5. Results of other pressure levels

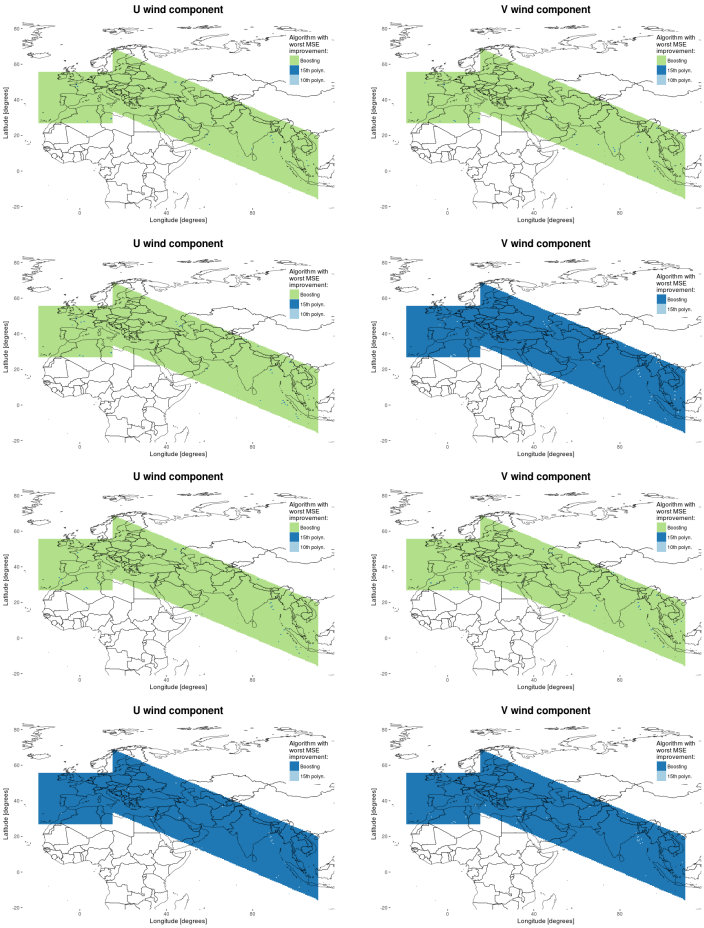


Figure A.12.: The worst performing algorithms. Illustrated are U and V wind components for 18 h (top row) through 00 h (bottom row) forecast steps, at 200 mbar altitude.

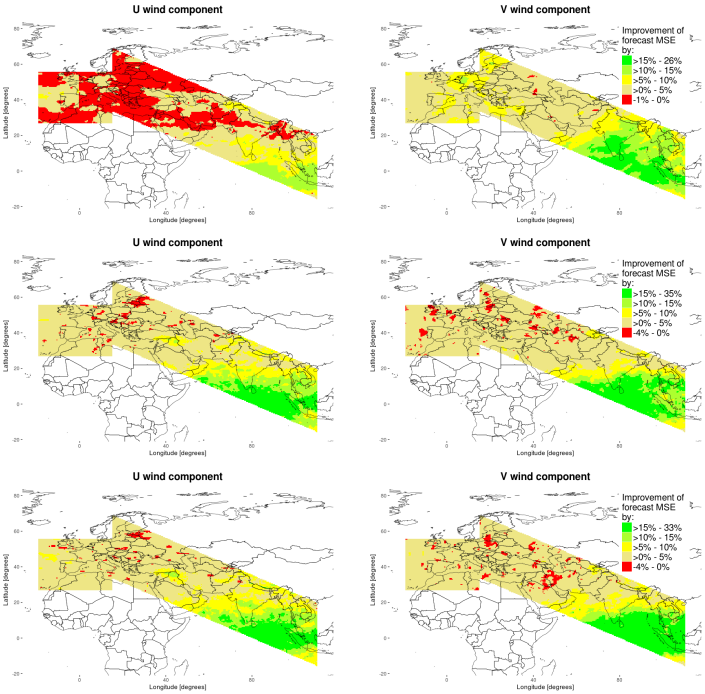


Figure A.13.: MSE test results for the 5th degree polynomial regression for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

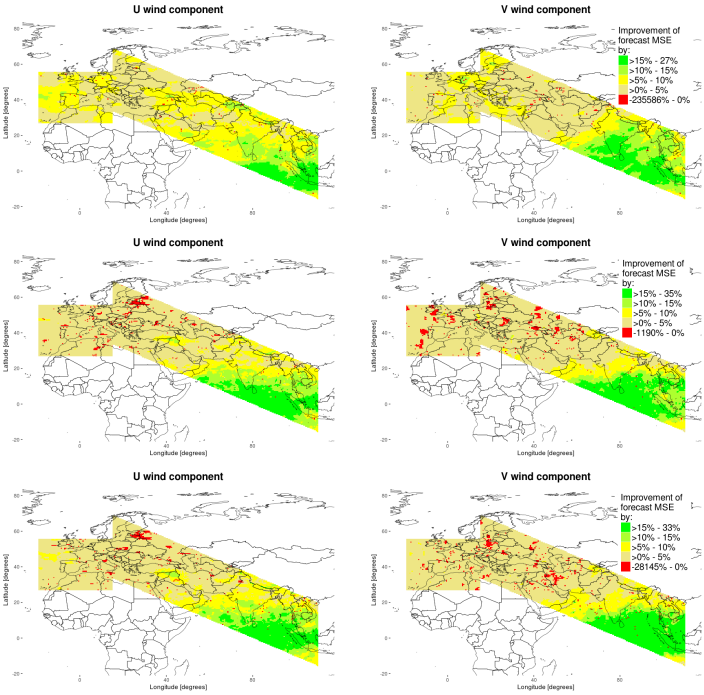


Figure A.14.: MSE test results for the 10th degree polynomial regression for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

A.6. Results from the algorithm selection method

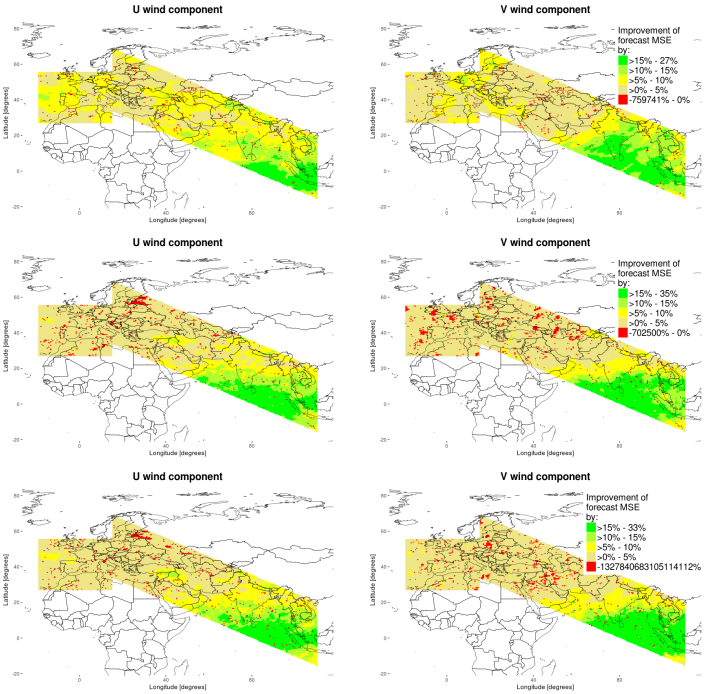


Figure A.15.: MSE test results for the 15th degree polynomial regression for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

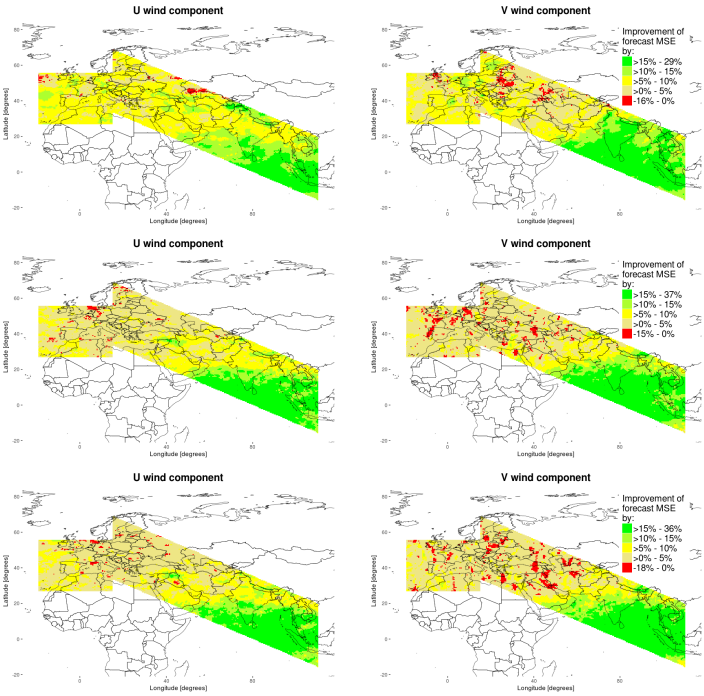


Figure A.16.: MSE test results for the SVM for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

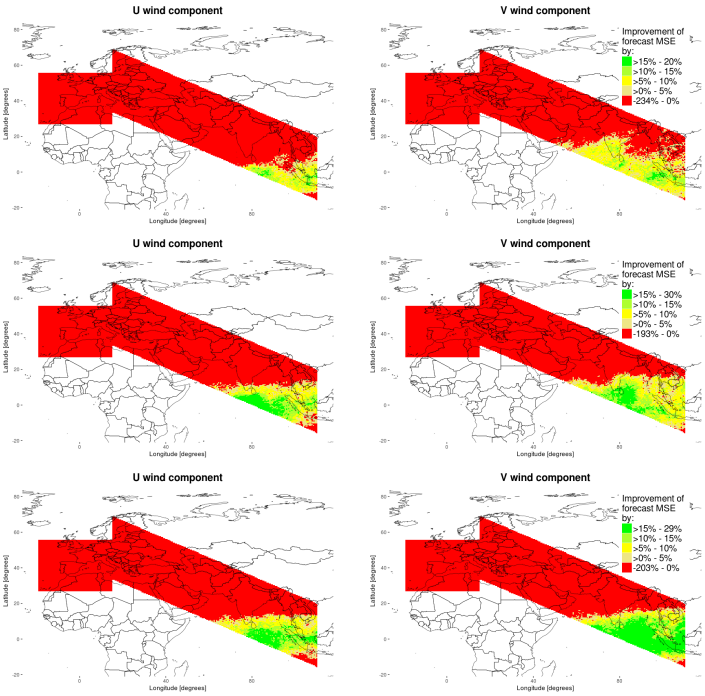


Figure A.17.: MSE test results for the decision tree for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

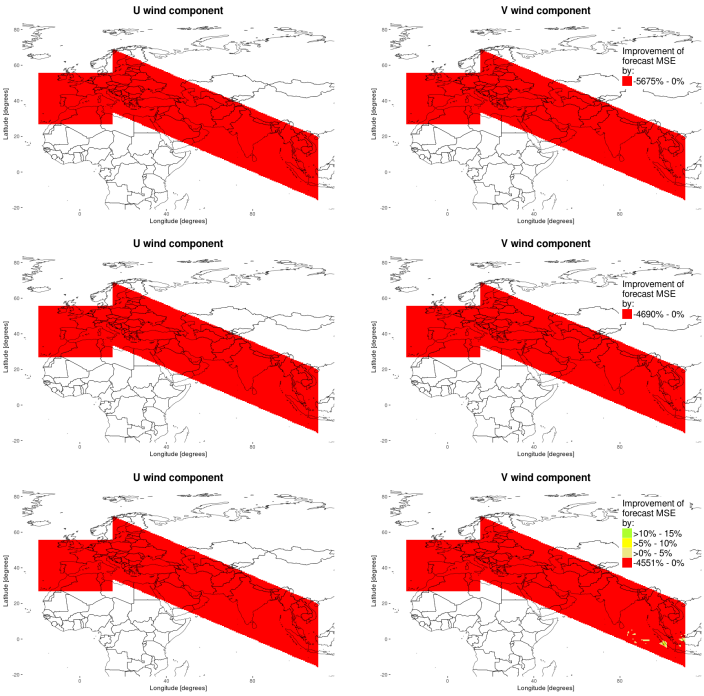


Figure A.18.: MSE test results for the Boosting algorithm for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

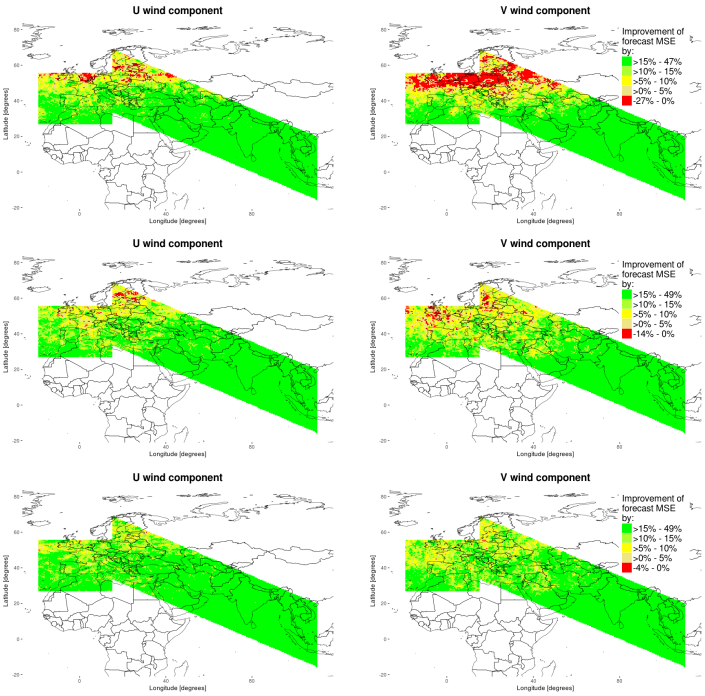


Figure A.19.: MSE test results for the kNN algorithm for U and V wind components at 150 mbar (top row), 250 mbar (center row) and 300 mbar (bottom row) altitude.

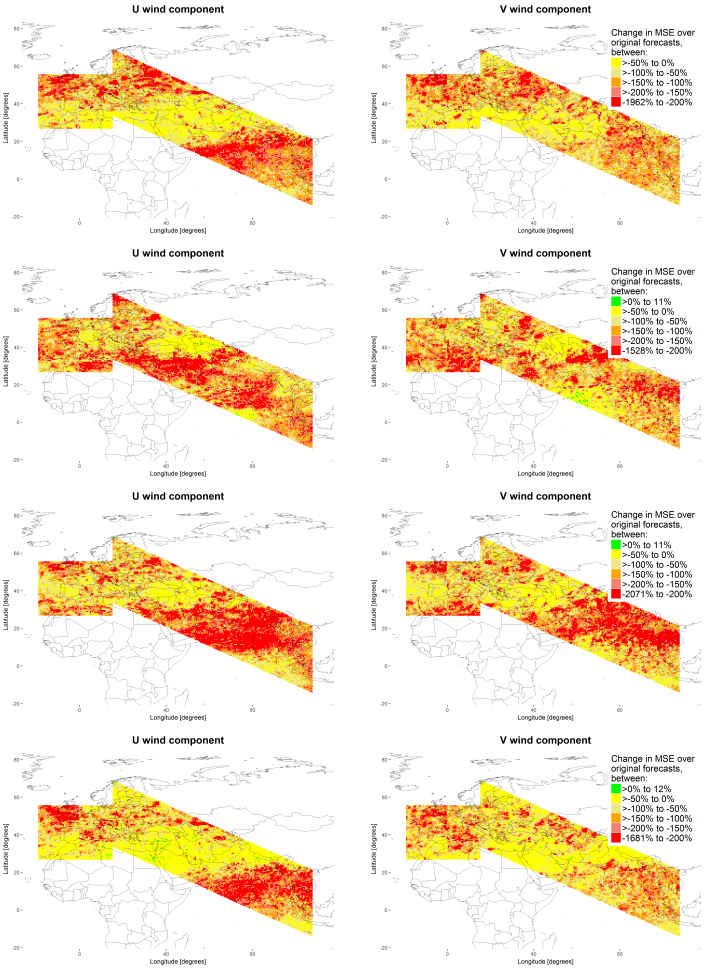


Figure A.20.: MSE performance results of the algorithm selection method across all four seasons, from Spring (top) to Winter (bottom), for 18 hour forecasts, at 200 mbar altitude, for U and V wind component.

A.7. Flight planning results

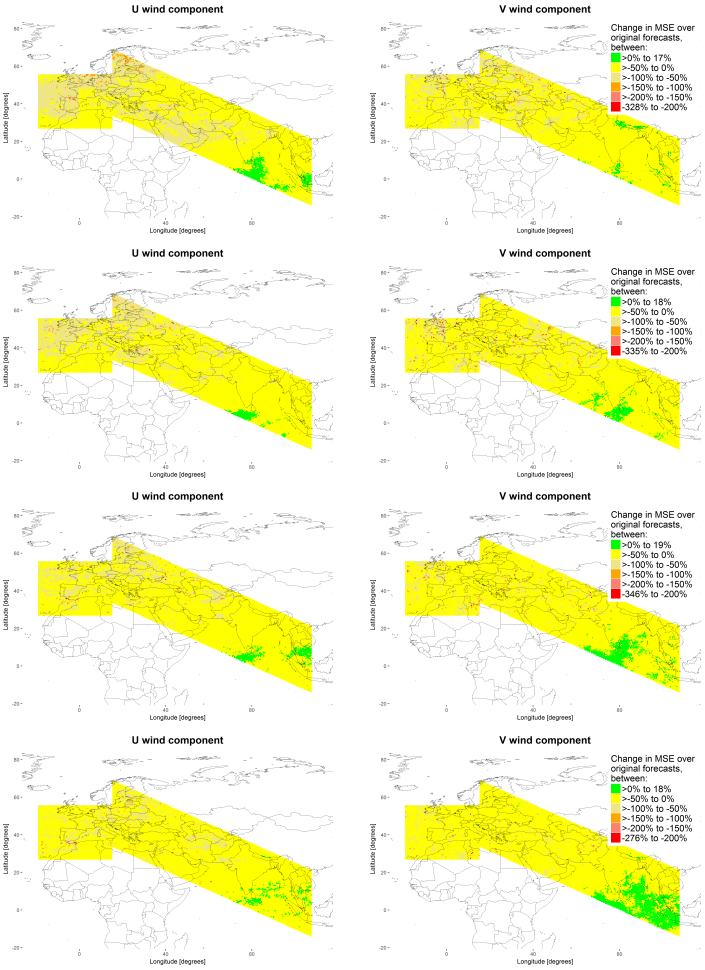


Figure A.21.: MSE performance results of the algorithm selection method across all four pressure levels, from 150 mbar (top) to 300 mbar (bottom) for 24 hour forecasts, for U and V wind components.

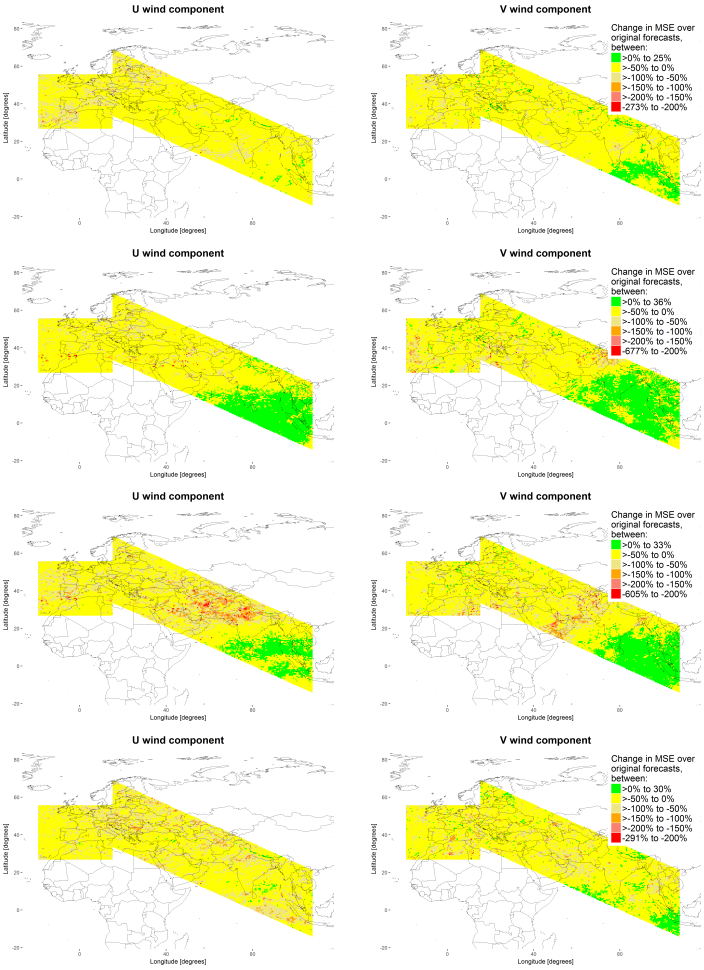


Figure A.22.: MSE performance results of the algorithm selection method across all four seasons, from spring (top) to winter (bottom) for 24 hour 300 mbar U and V wind component forecasts.

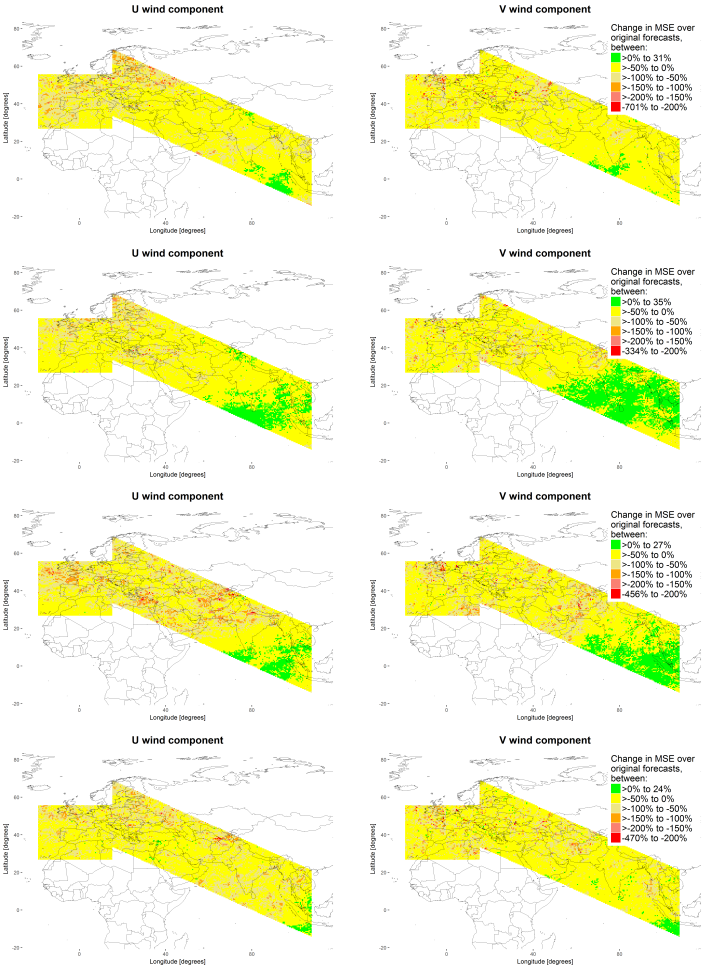


Figure A.23.: MSE performance results of the algorithm selection method across all four seasons, from spring (top) to winter (bottom), for 24 hour forecasts, at 200 mbar altitude, for U and V wind component.

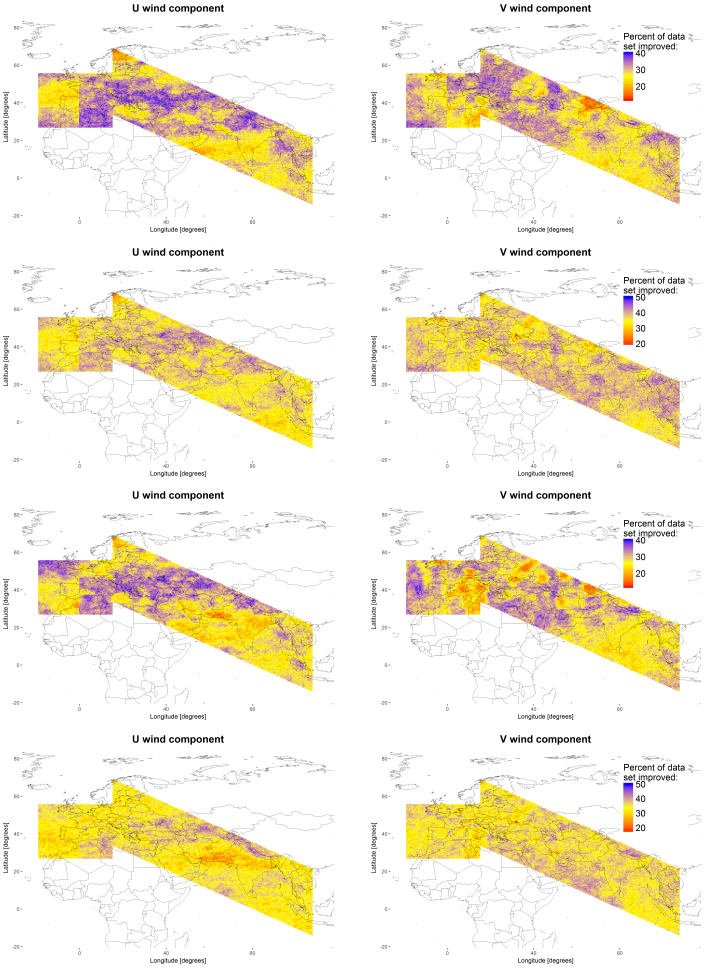


Figure A.24.: Percentage of data set improved in accuracy, at 200 mbar altitude, for 18 (top) to 00 (bottom) hour U and V wind component forecasts.

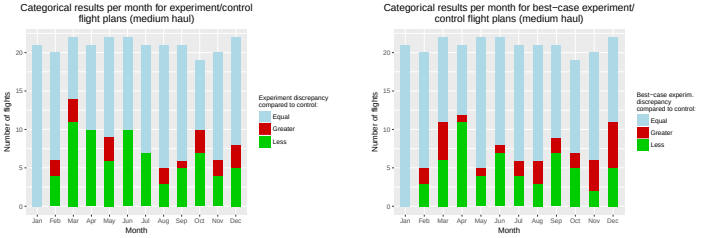


Figure A.25.: Monthly results for the experiment (left) and best-case experiment (right), for medium haul flight TUI2148.

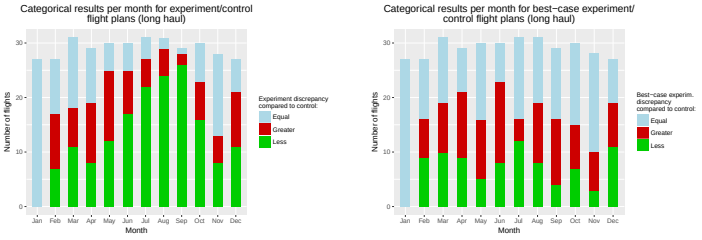


Figure A.26.: Monthly results for the experiment (left) and best-case experiment (right), for long haul flight DLH778.

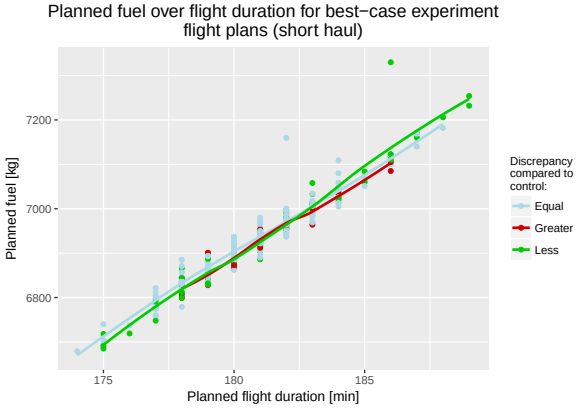


Figure A.27.: Best-case experiment: Planned fuel and flight time for short haul flight GIA867.

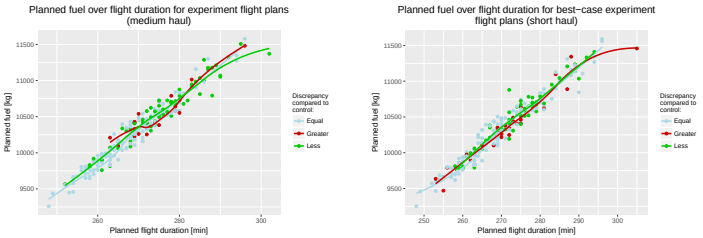


Figure A.28.: Planned fuel and flight time for medium haul flight TUI2148; experiment (left) and best-case experiment results (right).

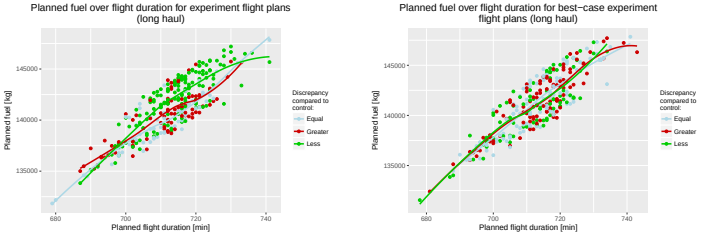


Figure A.29.: Planned fuel and flight time for long haul flight DLH778; experiment (left) and best-case experiment results (right).

List of Tables

List of Figures

List of Algorithms

Bibliography

[1] T. N. Palmer. Predicting uncertainty in forecasts of weather and climate. Technical memorandum no. 294, ECMWF, November 1999.

[2] Antonio S Cofiño, Rafael Cano, Carmen Sordo, and Jose M Gutierrez. Bayesian networks for probabilistic weather prediction. In *15th Eureopean Conference on Artificial Intelligence, ECAI*. Citeseer, 2002.

[3] Jacob Cheung, Jean-Louis Brenguier, Jaap Heijstek, Adri Marsman, and Helen Wells. Sensitivity of Flight Durations to Uncertainties in Numerical Weather Prediction. *Fourth SESAR Innovation Days, 25th-27th November 2014*, November 2014.

[4] Jacob Cheung, Alan Hally, Jaap Heijstek, Adri Marsman, and Jean-Louis Brenguier. Recommendations on trajectory selection in flight planning based on weather uncertainty. *Fifth SESAR Innovation Days, 1st-3rd December 2015*, December 2015.

[5] International Civil Aviation Organization. *Doc 9976. Flight Planning and Fuel Management Manual*. International Civil Aviation Organization, 2012.

[6] Rod Cole, Steve Green, Matt Jardin, Barry Schwartz, and Stan Benjamin. Wind prediction accuracy for air traffic management decision support tools. In *3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, Italy*. Europe/USA Air Traffic Management Research and Development Seminar, June 2000.

[7] Tom White. *Hadoop. The Definitive Guide*. O'Reilly Media, Inc., first edition edition, June 2009.

[8] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster Computing with Working Sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*. University of California, Berkeley, ACM, June 2010.

[9] Steve Altus. Effective Flight Plans Can Help Airlines Economize. *Boeing Aero Magazine*, 3:26–30, 2009.

[10] Stefan E. Karisch, Stephen S. Altus, Goran Stojković, and Mirela Stojković. *Quantitative Problem Solving Methods in the Airline Industry*, volume 169. Springer Science+Business Media, 2012.

[11] Lavanya Marla, Bo Vaaben, and Cynthia Barnhart. Integrated Disruption Management and Flight Planning to Trade off Delays and Fuel Burn. Research Paper, 2011.

[12] International Civil Aviation Organization. Annex 6, Operation of Aircraft. Part I, International Commercial Air Transport - Aeroplanes, July 2010.

[13] International Civil Aviation Organization. Procedures for Air Navigation Services - Air Traffic Management (PANS-ATM) Doc 4444, November 2007.

[14] Steve Altus. Flight Planning - the forgotten Field in Airline Operations `http://www.agifors.org/studygrp/opsctl/2007/`. In *Presented at AGIFORS Airline Operations*. AGIFORS, Jeppesen, May 2007.

[15] Steve Altus. Flight Planning Optimization Seminar. July 2015.

[16] Gerhard Brüning, Xaver Hafer, Gottfried Sachs, and Wolfgang Jurzig. *Flugleistungen - Grundlagen, Flugzustände, Flugabschnitte*, volume 3. Springer-Verlag, 1993.

[17] Carlos E. Padilla. *Optimizing Jet Transport Efficiency: Performance, Operations, and Economics*. McGraw-Hill Professional, 1 edition, July 1996.

[18] Airbus Customer Services. *Getting to grips with aircraft performance*. Airbus, January 2002.

[19] EUROCONTROL. *European Route Network Improvement Plan. PART 4, Route Availability Document Users Manual*. EUROCONTROL, 1.4 edition, June 2015.

[20] NCOIC. Comparison of the SESAR and NextGen Concepts of Operations, Version 1.0. 2008.

[21] H. M. de Jong. Optimal Track Selection and 3-dimensional Flight Planning. Research Report 93, Koninklijk Nederlands Meteorologisch Instituut, 1974.

[22] Ron McIntyre. The adverse impact of flight management systems on long range international airline operations. pages 359–363. IEEE, United Airlines, 1996.

[23] Arthur E. Bryson and Yu-Chi Ho. Applied optimal control: optimization, estimation, and control. Research paper, 1969.

[24] Arthur E. Bryson. Optimal Control - 1950 to 1985. *IEEE Control Systems*, pages 26–33, June 1996.

[25] R. E. Bellman. On a Routing Problem. *Quart. J. Appl. Math*, (16):87–90, 1958.

[26] Steve Altus. Dynamic Cost Index Management in Flight Planning and Replanning `http://www.agifors.org/studygrp/opsctl/2010/program.html`. In *Presented at AGIFORS Airline Operations*. AGIFORS, Jeppesen, May 2010.

[27] John A. Sorensen and Tsuyoshi Goka. Design of an advanced flight planning system. Research study paper, 1985.

[28] J. Wilson, C. Wright, and G. J. Couluris. Advanced Free Flight Planner and Dispatcher's Workstation. Technical report, Advanced Air Transportation Technology (AATT) Program, National Aeronautics and Space Administration, November 1997.

[29] Anandavel Murugan, Dinkar Mylaraswamy, Brian Xu, and Paul Dietrich. Big Data Infrastructure for Aviation Data Analytics. In *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, pages 1–6. IEEE, 2014.

[30] Rajendra Akerkar. Analytics on Big Aviation Data: Turning data into insights. In Prof. Avireni Srinivasulu, editor, *International Journal of Computer Science and Application*, volume 11. Technomathematics Research Foundation, 2014.

[31] Doug Laney. 3D data management: Controlling data volume, velocity and variety. Technical report, META Group Research Note, 2001.

[32] Tulinda Larsen. Cross-platform aviation analytics using big-data methods. In *Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 1–9. IEEE, April 2013.

[33] Kapil Bakshi. Considerations for big data: Architecture and approach. In *Aerospace Conference, 2012 IEEE*, pages 1–7. IEEE, 2012.

[34] Sameer Farooqi. Apache Hadoop: Devops Fundamentals. In *Big Data TechCon San Francisco*. Blue Plastic, October 2014.

[35] Dhruba Borthakur. The Hadoop Distributed File System: Architecture and Design. Technical report, The Apache Software Foundation, 2007.

[36] Jiong Xie, Shu Yin, Xiaojun Ruan, Zhiyang Ding, Yun Tian, James Majors, Adam Manzanares, and Xiao Qin. Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters. In *Proc. 19th International Heterogeneity in Computing Workshop*. Auburn University, IEEE, April 2010.

[37] Yanpei Chen, Sara Alspaugh, and Randy Katz. Interactive Analytical Processing in Big Data Systems: A Cross-Industry Study of MapReduce Workloads. In *Proceedings of the Very Large Data Bases Endowment (VLDB)*, volume 5, pages 1802–1813. University of California, Berkeley, VLDB Endowment, August 2012.

[38] Rajeev Gupta, Himanshu Gupta, and Mukesh Mohania. Cloud Computing and Big Data Analytics: What Is New from Databases Perspective? In Srinath Srinivasa and Vasudha Bhatnagar, editors, *First International Conference, Big Data Analytics*, 7678, pages 42–61. IBM Research India, Springer Berlin Heidelberg, December 2012.

[39] Brad Severtson. Introduction to Azure HDInsight. *Microsoft TechNet Articles*, (Online source, accessed on 14.09.2015) `http://social.technet.microsoft.com/wiki/contents/articles/13820.introduction-to-azure-hdinsight.aspx`, October 2012.

[40] Edmon Begoli and James Horey. Design principles for effective knowledge discovery from big data. In *2012 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) and European Conference on Software Architecture (ECSA)*, pages 215–218. IEEE, 2012.

[41] Erton Boci and Susan Thistlethwaite. A novel big data architecture in support of ADS-B data analytic. In *Integrated Communication, Navigation, and Surveillance Conference (ICNS), 2015*, pages C1–1–C1–8. IEEE, 2015.

[42] Serdal Ayhan, J Pesce, P Comitz, D Sweet, Steve Bliesner, and G Gerberick. Predictive analytics with aviation big data. In *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2013*, pages 1–13. IEEE, 2013.

[43] Abdul Ghaffar Shoro and Tariq Rahim Soomro. Big data analysis: Apache Spark perspective. *Global Journal of Computer Science and Technology*, 15(1), 2015.

[44] Ralf René Cabos, Peter Hecker, Nils Kneuper, and Jens Schiefele. Wind forecast uncertainty prediction using Machine Learning techniques on Big Weather Data. In *17th AIAA Aviation Technology, Integration, and Operations Conference*, page 3077, June 2017.

[45] Foster Provost and Tom Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking.* O'Reilly Media, Inc., 2013.

[46] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques: concepts and techniques.* Elsevier, 2012.

[47] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *KDD*, volume 96, pages 82–88, 1996.

[48] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–54, 1996.

[49] Pat Langley and Herbert A Simon. Applications of machine learning and rule induction. *Communications of the ACM*, 38(11):54–64, 1995.

[50] Joyce Jackson. Data Mining; A Conceptual Overview. *Communications of the Association for Information Systems*, 8(1):267–296, 2002.

[51] John E Laird, Paul S Rosenbloom, and Allen Newell. Chunking in Soar: The anatomy of a general learning mechanism. *Machine learning*, 1(1):11–46, 1986.

[52] BN Lakshmi and GH Raghunandhan. A conceptual overview of data mining. In *2011 National Conference on Innovations in Emerging Technology (NCOIET)*, pages 27–32. IEEE, 2011.

[53] Christopher M Bishop. Neural networks: a pattern recognition perspective. *Neural Computing Research Group*, 1996.

[54] Tom M Mitchell. *Machine Learning.* McGraw-Hill Boston, MA, 1997.

[55] Chris Chatfield. *Time-series forecasting.* Chapman and Hall/CRC, 2000.

[56] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[57] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[58] Marc Claesen, Frank De Smet, Johan AK Suykens, and Bart De Moor. Fast prediction with SVM models containing RBF kernels. *arXiv preprint arXiv:1403.0736*, 2014.

[59] Hui Cao, Takashi Naito, and Yoshiki Ninomiya. Approximate RBF kernel SVM and its applications in pedestrian classification. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis-MLVMA'08*, 2008.

[60] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[61] Jiang Su and Harry Zhang. A fast decision tree learning algorithm. In *AAAI*, volume 6, pages 500–505, 2006.

[62] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[63] Tom Armes and Mark Refern. Using Big Data and predictive machine learning in aerospace test environments. In *AUTOTESTCON*, pages 1–5. IEEE, 2013.

[64] Grunde Lovoll and Jürgen Christian Kadal. Big Data - the new data reality and industry impact. Technical report, DNV GL, April 2014.

[65] Jay Lee, Edzel Lapira, Behrad Bagheri, and Hung-an Kao. Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters*, 1(1):38–41, 2013.

[66] Airbus. Airbus E-solutions `http://www.airbus.com/support/flight-operations/e-solutions/`. *(Online source, last accessed on 11.02.16).*

[67] Lufthansa Systems. Lufthansa Industry Solutions `https://lufthansa-industry-solutions.de/technologien-services/big-data.html`. *(Online source, last accessed on 11.02.16).*

[68] General Electric. GE Digital Systems `http://www.geaviation.com/commercial/systems/digital-systems/`. *(Online source, last accessed on 11.02.16).*

[69] Bernard Marr. How Big Data Drives Success At Rolls-Royce. *Forbes Tech*, June 2015.

[70] Pratt & Whitney. Pratt & Whitney's 'Big Data' Projects Advancing Analytics Efforts in Aftermarket http://www.utc.com/News/PW/Pages/Pratt-Whitneys-Big-Data-Projects-Advancing-Analytics-Efforts-in-Aftermarket.aspx. *(Online source, last accessed on 11.02.16).*

[71] Boeing Edge. Airplane Health Management http://www.boeing.com/assets/pdf/commercial/aviationservices/brochures/AirplaneHealthManagement.pdf. *(Online source, last accessed on 11.02.16).*

[72] Ian T Jolliffe and David B Stephenson. *Forecast verification: a practitioner's guide in atmospheric science.* John Wiley & Sons, 2 edition, 2012.

[73] Patrick Hughes. The great leap forward. *Weatherwise*, 47(5):22–27, 1994.

[74] Allan H Murphy and Robert L Winkler. Probability forecasting in meteorology. *Journal of the American Statistical Association*, 79(387):489–500, 1984.

[75] Peter Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431–3444, 2008.

[76] Allan H Murphy. The value of climatological, categorical and probabilistic forecasts in the cost-loss ratio situation. *Monthly Weather Review*, 105(7):803–816, 1977.

[77] Lewis Fry Richardson. *Weather prediction by numerical process.* Cambridge University Press, 2007.

[78] G Marchuk. *Numerical methods in weather prediction.* Elsevier, 2012.

[79] Wolfgang Enke and Arne Spekat. Downscaling climate model outputs into local and regional weather elements by classification and regression. *Climate Research*, 8(3):195–207, 1997.

[80] Enrica Bellone, James P Hughes, and Peter Guttorp. A hidden Markov model for downscaling synoptic atmospheric patterns to precipitation amounts. *Climate research*, 15(1):1–12, 2000.

[81] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14):2627–2636, 1998.

[82] Benyang Tang and William W Hsieh. Applying neural network models to prediction and data analysis in meteorology and oceanography. *Atmospheric Science Program*, 1998.

[83] Robert J Kuligowski and Ana P Barros. Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. *Weather and Forecasting*, 13(4):1194–1204, 1998.

[84] Vladimir M Krasnopolsky and Michael S Fox-Rabinovitz. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, 19(2):122–134, 2006.

[85] Matthew de Kock, Hanh Le, Mark Tadross, and Anet Potgeiter. Weather Forecasting Using Dynamic Bayesian Networks. B.Sc. thesis, 2008.

[86] Ashish Kapoor, Zachary Horvitz, Spencer Laube, and Eric Horvitz. Airplanes aloft as a sensor network for wind forecasting. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 25–34. IEEE Press, 2014.

[87] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A Deep Hybrid Model for Weather Forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–386. ACM, 2015.

[88] John P Finley. Character of six hundred tornadoes. *US Signal Service Professional Paper*, 7:16, 1884.

[89] Allan H Murphy. The Finley affair: A signal event in the history of forecast verification. *Weather and Forecasting*, 11(1):3–20, 1996.

[90] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[91] Allan H Murphy. What is a good forecast? An essay on the nature of goodness in weather forecasting. *Weather and forecasting*, 8(2):281–293, 1993.

[92] Allan H Murphy and Robert L Winkler. A general framework for forecast verification. *Monthly Weather Review*, 115(7):1330–1338, 1987.

[93] Allan H Murphy. Forecast verification: Its complexity and dimensionality. *Monthly Weather Review*, 119(7):1590–1601, 1991.

[94] Henry R Stanski, Laurence J Wilson, and William R Burrows. *Survey of common verification methods in meteorology*. World Meteorological Organization Geneva, 1989.

[95] P Mailier, I Jollife, and D Stephenson. Quality of weather forecasts. *Review and recommendations Royal Meteorological Society*, pages 1–89, 2006.

[96] Pascal J Mailier, Ian T Jolliffe, and David B Stephenson. Assessing and reporting the quality of commercial weather forecasts. *Meteorological Applications*, 15(4):423–429, 2008.

[97] National Centers for Environment Prediction. Global Forecast System (GFS) `https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs`. *(Online source, last accessed on 11.05.16)*, 2016.

[98] Grace Peng. What's the difference between FNL and GFS? *Atmospheric & Geoscience Research Data Archive, Computational & Information Systems Laboratory, National Center for Atmospheric Research*, December 2014.

[99] Grace Peng. Analysis, reanalysis, forecast - what's the difference? *Atmospheric & Geoscience Research Data Archive, Computational & Information Systems Laboratory, National Center for Atmospheric Research*, December 2014.

[100] Suranjana Saha, Shrinivas Moorthi, Hua-Lu Pan, Xingren Wu, Jiande Wang, Sudhir Nadiga, Patrick Tripp, Robert Kistler, John Woollen, David Behringer, et al. The NCEP climate forecast system reanalysis. *Bulletin of the American Meteorological Society*, 91(8):1015–1057, January 2010.

[101] DP Dee, SM Uppala, AJ Simmons, Paul Berrisford, P Poli, S Kobayashi, U Andrae, MA Balmaseda, G Balsamo, P Bauer, et al. The ERA-Interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137(656):553–597, April 2011.

[102] Eugenia Kalnay, Masao Kanamitsu, Robert Kistler, William Collins, Dennis Deaven, Lev Gandin, Mark Iredell, Suranjana Saha, Glenn White, John Woollen, et al. The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–471, 1996.

[103] Unidata, University Corporation for Atmospheric Research and National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce and European Centre for Medium-Range Weather Forecasts. Historical Unidata Internet Data Distribution (IDD) Gridded Model Data. Last accessed 18-07-2016, 2003.

[104] Julian M. Wright. *Federal meteorological handbook No. 3: Rawinsonde and Pibal Obervations*. Office of the Federal Coordinator for Meteorological Services and Supporting Research, Silver Spring, Maryland 20910, May 1997.

[105] WM Mularie. Department of defense world geodetic system 1984, its definition and relationships with local geodetic systems. Technical report, National Imagery and Mapping Agency - Department of Defense, January 2000.

[106] Arnab Nilim, Laurent El Ghaoui, and Vu Duong. Multi-aircraft routing and traffic flow management under uncertainty. In *5th USA/Europe Air Traffic Management Research and Development Seminar, Budapest, Hungary*, pages 23–27, 2003.

[107] Manuela Sauer, Thomas Hauf, and Caroline Forster. Uncertainty Analysis of Thunderstorm Nowcasts for Utilization in Aircraft Routing. *Fourth SESAR Innovation Days*, 2014.

[108] Thomas Hauf, Ludmila Sakiew, and Manuela Sauer. Adverse weather diversion model DIVMET. *Journal of Aerospace Operations*, 2(3-4):115–133, 2013.

[109] Patrick Hupe, Thomas Hauf, and Carl-Herbert Rokitansky. Case Study of Adverse Weather Avoidance Modelling. *Fourth SESAR Innovation Days*, November 2014.

[110] Manuela Sauer, Ludmila Sakiew, Thomas Hauf, and Patrick Hupe. 8.6 Some Applications of the adverse weather diversion model DIVMET. 2013.

[111] Manuela Sauer. DIVMET - Modellierung von Flugroutenänderungen auf Grund von Gewittern. *(Online source, last accessed on 11.02.2016)* `http://www.muk.uni-hannover.de/309.html`, 2014.

[112] Christina Schilke and Peter Hecker. Dynamic Route Optimization Based on Adverse Weather Data. *Fourth SESAR Innovation Days*, 2014.

[113] Clayton P Tino, Liling Ren, and John-Paul B Clarke. Wind forecast error and trajectory prediction for en-route scheduling. In *AIAA Guidance, Navigation, and Control Conference, Chicago*, 2009.

[114] Clayton P Tino. *Wind models and stochastic programming algorithms for en route trajectory prediction and control*. PhD thesis, Georgia Institute of Technology, 2013.

[115] Roberto Buizza, M Milleer, and TN Palmer. Stochastic representation of model uncertainties in the ECMWF ensemble prediction system. *Quarterly Journal of the Royal Meteorological Society*, 125(560):2887–2908, 1999.

[116] G Candille, C Côté, PL Houtekamer, and G Pellerin. Verification of an ensemble prediction system against observations. *Monthly Weather Review*, 135(7):2688–2699, 2007.

[117] ComplexWorld.eu. Complexity Challenges in ATM. Summary document, Innaxis, May 2016.

[118] John Gulding, David Knorr, Marc Rose, James Bonn, Philippe Enaud, and Holger Hegendoerfer. US/Europe comparison of ATM-related operational performance. *Europe*, 4(6):8, March 2010.

[119] Barry E Schwartz, Stanley G Benjamin, Steven M Green, and Matthew R Jardin. Accuracy of RUC-1 and RUC-2 wind and aircraft trajectory forecasts by comparison with ACARS observations. *Weather and Forecasting*, 15(3):313–326, 2000.

[120] Alan G Lee, Stephen S Weygandt, Barry Schwartz, and James R Murphy. Performance of trajectory models with wind uncertainty. In *AIAA Modeling and Simulation Technologies Conference, Chicago, Illinois*. American Institute of Aeronautics and Astronautics, 2009.

[121] Q Maggie Zheng and JY Zhao. Modeling wind uncertainties for stochastic trajectory synthesis. In *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, pages 20–22. American Institute of Aeronautics and Astronautics, September 2011.

[122] Jonathan D Kahl and Perry J Samson. Uncertainty in trajectory calculations due to low resolution meteorological data. *Journal of climate and applied meteorology*, 25(12):1816–1831, 1986.

[123] Stéphane Mondoloni. A multiple-scale model of wind-prediction uncertainty and application to trajectory prediction. In *6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, pages 1–14. American Institute of Aeronautics and Astronautics, 2006.

[124] Joshua W Pepper, Kristine R Mills, and Leonard A Wojcik. Predictability and uncertainty in air traffic flow management. In *5th USA/Europe Air Traffic Management R&D Seminar (ATM-2003), Metrics and Performance Management,*

Budapest, Hungary. Europe/USA Air Traffic Management Research and Development Seminar, June 2003.

[125] John-Paul B Clarke, Senay Solak, Yu-Heng Chang, Liling Ren, and Adan E Vela. Air traffic flow management in the presence of uncertainty. In *Proceedings of the 8th USA/Europe Air Traffic Seminar (ATM'09)*. Europe/USA Air Traffic Management Research and Development Seminar (ATM 2009), 2009.

[126] World Meteorological Organization. *Introduction to GRIB Edition1 and GRIB Edition 2*. World Meteorological Organization, June 2003.

[127] Jeppesen. *JetPlan User Manual*. Jeppesen - A Boeing Company, 7.0 edition, February 2015.

[128] International Civil Aviation Organization. *Doc 7488/3: Manual of the ICAO Standard Atmosphere extended to 80 kilometres (262500 feet)*. ICAO, 3 edition, 1993.

[129] Federal Aviation Administration. *Aeronautical Information Manual*. U.S. Department of Transportation, 800 Independence Ave., S.W. Washington, D.C. 20591, May 2016.

[130] Mark Chong. The role of internal communication and training in infusing corporate values and delivering brand promise: Singapore Airlines' experience. *Corporate Reputation Review*, 10(3):201–212, September 2007.

[131] FlightAware. Singapore Airlines 22, SIA22. Website, last accessed on 03-08-2016, November 2013.

[132] Suranjana Saha, Shrinivas Moorthi, Hua-Lu Pan, Xingren Wu, Jie Wang, Sudhir Nadiga, Patrick Tripp, Robert Kistler, John Woollen, David Behringer, Haixia Liu, Diane Stokes, Robert Grumbine, George Gayno, Jun Wang, Yu-Tai Hou, Hui-Ya Chuang, Hann-Ming H. Juang, Joe Sela, Mark Iredell, Russ Treadon, Daryl Kleist, Paul Van Delst, Dennis Keyser, John Derber, Michael Ek, Jesse Meng, Helin Wei, Rongqian Yang, Stephen Lord, Huug van den Dool, Arun Kumar, Wanqiu Wang, Craig Long, Muthuvel Chelliah, Yan Xue, Boyin Huang, Jae-Kyung Schemm, Wesley Ebisuzaki, Roger Lin, Pingping Xie, Mingyue Chen, Shuntai Zhou, Wayne Higgins, Cheng-Zhi Zou, Quanhua Liu, Yong Chen, Yong Han, Lidia Cucurull, Richard W. Reynolds, Glenn Rutledge, and Mitch Goldberg. NCEP Climate Forecast System Reanalysis (CFSR) 6-hourly Products, January 1979 to December 2010. Last accessed 18-07-2016, 2010.

[133] Suranjana Saha, Shrinivas Moorthi, Xingren Wu, Jiande Wang, Sudhir Nadiga, Patrick Tripp, David Behringer, Yu-Tai Hou, Hui ya Chuang, Mark Iredell, Michael Ek, Jesse Meng, Rongqian Yang, Malaquias Pena Mendez, Huug van den Dool, Qin Zhang, Wanqiu Wang, Mingyue Chen, and Emily Becker. NCEP Climate Forecast System Version 2 (CFSv2) 6-hourly Products. Last accessed 18-07-2016, 2011.

[134] Suranjana Saha, Shrinivas Moorthi, Xingren Wu, Jiande Wang, Sudhir Nadiga, Patrick Tripp, David Behringer, Yu-Tai Hou, Hui-ya Chuang, Mark Iredell, et al. The NCEP climate forecast system version 2. *Journal of Climate*, 27(6):2185–2208, March 2014.

[135] Lizhe Wang, Jie Tao, Holger Marten, Achim Streit, Samee U Khan, Joanna Kolodziej, and Dan Chen. MapReduce across distributed clusters for data-intensive applications. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, pages 2004–2011. IEEE, May 2012.

[136] RStudio. *RStudio Server Administrator's Guide*. RStudio Inc., 2013.

[137] Kay Ousterhout, Ryan Rasti, Sylvia Ratnasamy, Scott Shenker, and Byung-Gon Chun. Making sense of performance in data analytics frameworks. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 293–307, May 2015.

[138] Avrilia Floratou, Umar Farooq Minhas, and Fatma Özcan. SQL-on-Hadoop: Full circle back to shared-nothing database architectures. In *Proceedings of the VLDB Endowment*, number 12, pages 1295–1306. VLDB Endowment, 2014.

[139] Hadley Wickham. Tidy Data. *Journal of Statistical Software*, 59(10):1–23, August 2014.

[140] E Dimitriadou, K Hornik, F Leisch, D Meyer, and Weingessel A. Machine Learning Open-Source Package'r-cran-e1071' [R package Version 1.6-8. 2017.

[141] Ivor W Tsang, James T Kwok, and Pak-Ming Cheung. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6(Apr):363–392, 2005.

[142] B.D. Ripley. Package 'tree' [R package Version 1.0-37]. 2016.

[143] J Kent Martin and DS Hirschberg. On the complexity of learning decision trees. In *International Symposium on Artificial Intelligence and Mathematics*, pages 112–115, 1996.

[144] Greg Ridgeway. Package 'gbm' [R package Version 2.1.3]. 2017.

[145] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[146] J Shukla and Daniel A Paolino. The Southern Oscillation and long-range forecasting of the summer monsoon rainfall over India. *Monthly Weather Review*, 111(9):1830–1837, 1983.

[147] TN Krishnamurti, CM Kishtawal, Timothy E LaRow, David R Bachiochi, Zhan Zhang, C Eric Williford, Sulochana Gadgil, and Sajani Surendran. Improved weather and seasonal climate forecasts from multimodel superensemble. *Science*, 285(5433):1548–1550, 1999.

[148] Nikolaos Papadakos. Integrated airline scheduling. *Computers & Operations Research*, 36(1):176–195, 2009.

[149] Christian Haude. Development of an Analysis System for Deviation Calculation Between Scheduled Flight Plans and Real Flight Trajectories. Master's thesis, Institut für Flugführung, Technische Universität Braunschweig, 2017.

Niedersächsisches Forschungszentrum für Luftfahrt
Hermann-Blenk-Straße 27
38108 Braunschweig
Telefon +49 531 391-9822
Telefax +49 531 391-9804
nfl@tu-braunschweig.de
www.nfl.tu-braunschweig.de

ISBN 978-3-947623-04-4